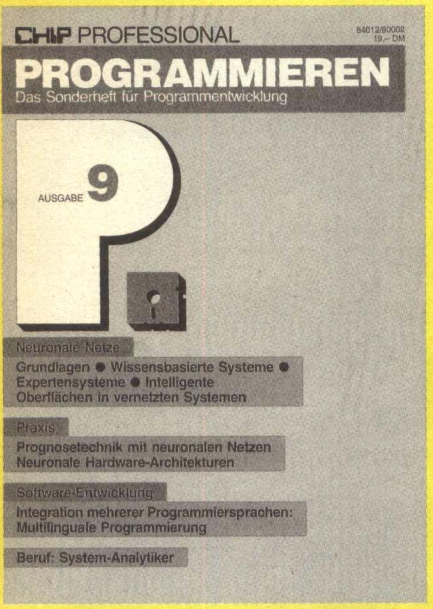
CHIP PLUS

EINE BEILAGE
IN CHIP NR. 5
MAI 1990

68040-MIKROPROZESSOR

PROGRAMMIERWISSEN FÜR ALLE

Programmierwissen wird in der Reihe CHIP PROFESSIONAL PROGRAMMIEREN so aufbereitet, daß alle Programm-Entwickler davon profitieren — unabhängig vom Betriebssystem und von der Programmiersprache. Flächenübergreifende Themen aus dem oft facettenreichen Alltag der Programmier-Berufe sind Mittelpunkt der Beiträge, die von Profis aus allen Disziplinen, die für Tätigkeiten am und mit dem Computer relevant sind, aufgegriffen und diskutiert werden.
Jede Ausgabe behandelt ein Schwerpunktthema. Diese vermitteln nicht nur Grundlagen, sondern auch praktische Einsatzgebiete.



NEU

Programmieren 9
Die Vorteile neuronaler Netze, dem Schwerpunktthema dieser Ausgabe, lassen sich mit Hilfe des notwendigen Basiswissens auf sequentiell arbeitenden Computern simulieren. In einführenden Beiträgen wird erklärt, wie die Topologie des gewünschten Netzes definiert und wie – auf der Basis eher einfacher Regeln – bestimmte Erregungsmuster mit Hilfe wissensbasierter Systeme „erlernt" werden, damit das System künftige Fragen zu beantworten vermag. Aus dem Inhalt: Intelligente Oberflächen in vernetzten Systemen; Prognosetechnik mit neuronalen Netzen; Neuronale Hardware-Architektur; Multilinguale Programmierung.
Best.-Nr.: 1060, DM 19,-

Programmieren 8
„Computergrafik und Animation" ist der Schwerpunkt dieser Ausgabe, in der die Methoden, Techniken, Strukturen, Programm-Entwicklung und Anwendungen aufgezeigt werden. Wie sich das Arbeiten mit Grafik-Applikationen gestaltet zeigt das Praxis-Beispiel. Die Compilerbau-Serie schließt mit den Themen Codeoptimierung und Datenstrukturen ab. Exklusiv: Gespräch mit Konrad Zuse. Berufsthema: Animation Designer.
Best.-Nr.: 1004, DM 19,-

Programmieren 7
Schwerpunktthema: Objektorientiertes Programmieren. Grundlagen, Abstrakte Datentypen, Objekte in Programmiersprachen und OOP als Programmierwerkzeug Praktikum: Splines — dem elektronischen Kurvenlineal. Compilerbau: Parser und Codegenerator. Berufsbild: Informatiker. Exklusivinterview mit Professor Josef Weizenbaum.
Best.-Nr.: 0996, DM 19,-,

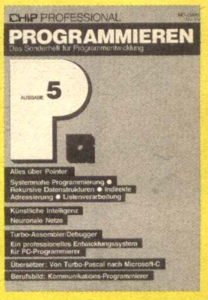
Programmieren 6
Schwerpunktthema: Compilerbau. Grundlagen ‚Formale Sprachen, Scanner, Parser, Codegenerator, Praktische Ausführung. Praktikum: Fast-Fourier-Transformation, Digitale Filter. Supercomputer: Die Programmierung des Suprenum-Rechners, Nachdokumentations-Werkzeuge. Berufsbild: Organisationsprogrammier.
Best.-Nr.: 0993, DM 19,-,

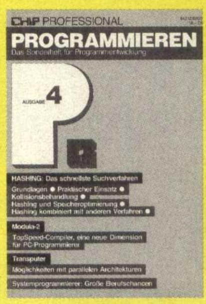
Programmieren 5
Schwerpunktthema: Alles über Pointer — Systemnahe Programmierung; Rekursive Datenstrukturen; Indirekte Adressierung. Inhalt: Künstliche Intelligenz — Neuronale Netze; Superrechner — Suprenum; Turbo-Assembler/Debugger — ein Entwicklungssystem für PC-Programmierer; Übersetzer von Turbo-Pascal nach Microsoft C; Berufsbild: Kommunikationsprogrammierer.
Best.-Nr.: 0990, DM 19,-,

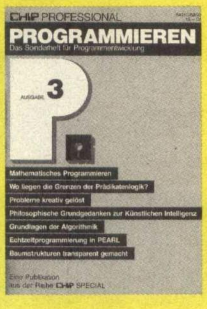




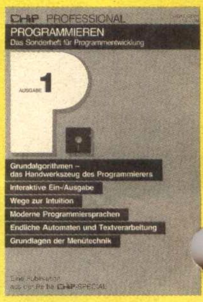












Programmieren 4
Schwerpunktthema: Hashing. Das schnellste Suchverfahren; Grundlagen — praktischer Einsatz; Kollisionsbehandlung; Hashing und Speicheroptimierung; Hashing kombiniert mit anderen Verfahren. Inhalt: Modula-2 Top-Speed-Compiler — eine neue Dimension für PC-Programmierer; Transputer — Möglichkeiten mit parallelen Architekturen; Berufsbild: Systemprogrammierer. Nobelpreisträger Prof. Gerd Binnig: Die fraktale Struktur der Evolution.
Best.-Nr.: 0987, DM 19,-,

Programmieren 3
Aus dem Inhalt: Künstliche Intelligenz im Brennpunkt; Modula 2 und C im Vergleich; Prädikatenlogik und ihre Grenzen; Stand der automatischen Fremdsprachen-Übersetzer; Die Echtzeitsprache Pearl; Grenzen der Programmierbarkeit.
Best.-Nr.: 0974, DM 19,-,

Programmieren 2
Aus dem Inhalt: Die Simplex-Methode — Suchverfahren und Datenstrukturen; Die Programmiersprachen der künstlichen Intelligenz; Grundlagen der Rekursion; Programmieren als Selbstzweck; Pointer — das Schreckgespenst vieler Programmierer.
Best.-Nr.: 0971, DM 19,-,

Programmieren 1
Aus dem Inhalt: Grundalgorithmen — das Handwerkszeug des Programmierers; Interaktive Ein- und Ausgabe; Wege zur Intuition; Moderne Programmiersprachen; Endliche Automaten und Textverarbeitung; Grundlagen der Menütechnik.
Best.-Nr.: 0965, DM 19,-,

COMPUTERWISSEN AUS DEM VOGEL VERLAG

Erhältlich beim ausgewählten Buch- und PC-Fachhandel

VOGEL
**Vogel Verlag
Postfach 6740
D-8700 Würzburg 1
Tel. 09 31/4 18-22 8**

Der Nachzügler ist da

Ulrich Parthier

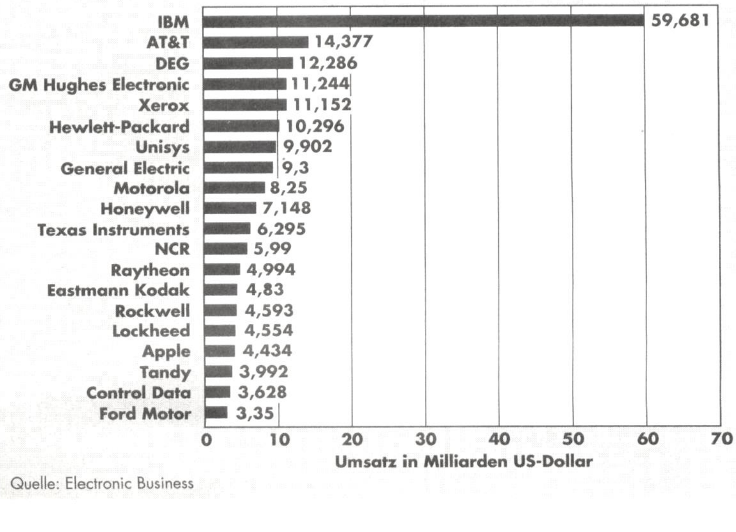
Nachdem Intel im April vergangenen Jahres sein neues Flaggschiff, den i486-Mikroprozessor, vorstellte, hat nun Erzrivale Motorola gleichgezogen. Im Januar dieses Jahres wurde nach längeren Verzögerungen nun endlich der 68040 offiziell vorgestellt – mit Leistungen, die ihn zum ernstzunehmenden Konkurrenten machen.

Gespannt sein darf man, wieviel Zeit bis zur Serienreife des 68040, das heißt der Verfügbarkeit in großen Stückzahlen, vergehen wird. Schließlich betrug diese Zeitspanne beim Konkurrenten i486 – nicht zuletzt wegen einiger Fehler – fast ein Jahr. Anscheinend läßt sich dies bei solch komplexen Chips trotz noch so langer Testverfahren nicht ganz ausschließen.

Um die Funktionen und Leistungen des neuen Motorola-Prozessors soll es in diesem CHIP PLUS gehen. Auch dem Intel-Flaggschiff hatten wir ein ganzes Heft gewidmet – in der September-Ausgabe des vergangenen Jahres. Auf einen Vergleich der beiden Chips wollen wir an dieser Stelle nicht eingehen. Über das Schicksal des Motorola-Mikroprozessors kann einzig und allein der Markt entscheiden.

An schnellere Prozessoren müssen wir uns gewöhnen. Auch bekommt das Silizium als Baustoff dieser Schaltkreise immer mehr Konkurrenz. Zahlreiche Projekte über Supraleiter, Gallium-Arsenid-Verbindungen oder in der Optoelektronik sind schon am Ende ihrer Experimentier- und Laborphase angelangt. Wann hier die ersten Produkte auf den Markt kommen, ist nur noch eine Frage der Zeit. Ob wir sie wirklich brauchen, ist eine zweite.

Wie enorm sich die Rechnerleistungen verändern werden, sei am Beispiel eines neuen, jetzt von den berühmten AT&T Bell Laboratories vorgestellten optischen Computers demonstriert. Bei diesem System wird ein Prozessor eingesetzt, der Laserlicht für die interne Übertragung und Weiterverarbeitung einsetzt. Dadurch ist er im Vergleich zu einem herkömmlichen Elektronenrechner um den Faktor tausend schneller. Der Prototyp kommt derzeit auf eine Leistung von eine Million Impulse pro Sekunde. Die Forscher sind sich jedoch sicher, daß sich die Taktrate auf mehrere hundert Millionen Zyklen pro Sekunde steigern läßt. Werden etwa Rechner mit dem i486 oder dem 68040 in wenigen Jahren nur noch für Low-End-Anwendungen wie Textverarbeitung oder Datenbanken eingesetzt? Wir werden es erleben.

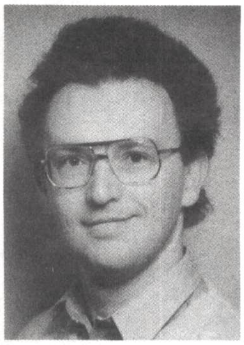


Quelle: Electronic Business

Mit dem 68040-Mikroprozessor will Motorola seine Stellung unter den 20 führenden US-Elektronik-Firmen halten

Der MC68040 – fix wie der schnellste RISC

Josef Fuchs



Der Autor Dipl.-Informatiker Josef Fuchs stammt aus Bogen/Niederbayern. Er studierte an der Technischen Universität München Informatik, Fachrichtung Elektrotechnik. Seit 1986 ist er bei Motorola in München als Field Applications Engineer für die 32-bit-Familie tätig.

Zwei Dinge sind es vor allem, die von den Mikroprozessor-Anwendern immer wieder gefordert werden: mehr Leistung und Kompatibilität zu den Vorgängern. Den Ruf nach mehr Leistung hat Motorola bereits mit der M88000-RISC-Familie erfüllt. Mit dem MC68040 kann nun eine ähnlich hohe Leistungsfähigkeit erreicht werden, wie sie ein MC88100 bereits bietet. In diesem Fall ist aber volle Software-Kompatibilität zur M68000-Familie gewährleistet.

Die technischen Daten des MC68040 im Kurzüberblick:

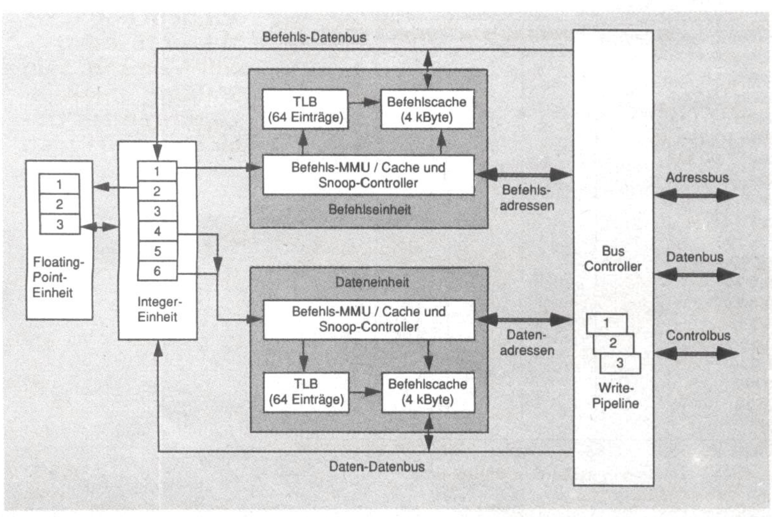
- volle Kompatibilität zu den übrigen Mikroprozessoren der M68000-Familie
- Subset des Coprozessors MC 68881/MC68882 on chip
- mit Floating-Point-Software-Paket vollständiger Funktionsumfang des MC 68881/882 bei voller Kompatibilität
- 3,6fache Integer-Leistung verglichen mit einem MC 68020 bei gleicher Taktfrequenz (Durchschnittswert)
- bis zu 30fache Floating-Point-Leistung verglichen mit dem MC 68882
- Floating-Point-Hardware-Multiplizierer
- Aufteilung in sieben unabhängige Blöcke, die parallel arbeiten können
- zusätzliche Pipelines in den Verarbeitungseinheiten
- zwei getrennte, physikalische Caches mit einer Größe von jeweils 4 KByte
- zwei PMMU für gleichzeitigen Zugriff auf die beiden physikalischen Caches
- Multiprozessorunterstützung durch Snoop-Logik

Der MC 68040 besteht aus fünf Blöcken, die weitgehend unabhängig voneinander parallel arbeiten können. Die Integer- und Floating-Point-Einheiten sind die eigentlichen Arbeitsbereiche. Sie enthalten eigene Pipelines für maximale Leistung. Die Befehlseinheit enthält Befehls-MMU mit Übersetzungs-Cache, Befehlscache und Snoop-Logik; sie versorgt die beiden Verarbeitungseinheiten mit den Befehlen und entlastet dabei gleichzeitig den Bus. Die Dateneinheit enthält Daten-MMU mit Übersetzungs-Cache, Datencache und Snoop-Logik. Sie dient dazu, die beiden Verarbeitungseinheiten mit Operanden zu versorgen und dabei den Bus möglichst gering in Anspruch zu nehmen. Der Buscontroller stellt die Schnittstelle zum externen Bus dar. Durch mehrere 128-bit-Register und eine dreistufige Schreib-Pipeline wird der externe Bus optimal genutzt. Für schnellstmöglichen Datentransfer sind ein Teil der internen Busse 128 bit breit.

Kombination aus Mikroprogramm und Hardware

Die M68000-Familie ist mikroprogrammiert. Das erlaubt sehr leistungsfähige Befehle und Adressierungsarten sowie zugleich einen übersichtlichen, modularen Aufbau des Bausteins. Die RISC-Familie versucht einen anderen Weg: Direkte Ausführung der einzelnen Befehle in Hardware. Das steigert zwar die Anzahl der Befehle pro Zeiteinheit, beschränkt aber gleichzeitig den Umfang und die Leistungsfähigkeit der einzelnen Befehle.

Der MC68040 versucht, die Vorteile der beiden Verfahren zu verbinden. Er ist zwar mikroprogrammiert, aber einzelne Schritte innerhalb vieler Befehle wurden vom Mikroprogramm in die Hardware verlagert. Ein Beispiel hierfür sind vor allem die Adressierungsarten. Die M68000-Serie besitzt eine sehr große Zahl von leistungsfähigen Adressierungsarten, die nur durch den Mikrocode möglich sind. Es wurden Untersuchungen angestellt, welche Adressierungsarten im Durchschnitt wie oft verwendet werden. Ein speziell für den MC68040 entwickelter Hardware-Decoder verarbeitet die am häufigsten verwendeten Adressierungsarten in Hardware (bei allen Vorgängern wurde das vom Mikrocode



Blockschaltbild des MC68040. Der Baustein besteht aus fünf weitgehend unabhängigen Blöcken, die parallel arbeiten können.

durchgeführt). Auch die Arbeit der beiden MMU beim MC 68030 durch den Mikrocode erledigt – wird nun in Hardware durchgeführt, wodurch die Geschwindigkeit der Adreßübersetzung drastisch gesteigert wird. Man beschränkte sich dabei auf die Tabellenstrukturen, die in der Praxis fast ausschließlich benutzt werden.

Integer-Einheit

Die Integer-Einheit ist funktions- und objectcode-kompatibel zu der des MC68020/68030. Viele Optimierungen sorgen dafür, daß deren Leistung im Durchschnitt um den Faktor 3,6 gegenüber dem MC68020 bei gleicher Taktfrequenz (BCLOCK) gesteigert werden konnte. Dieser Faktor setzt sich aus mehreren Dingen zusammen: Der größte Leistungsschub wurde dadurch erreicht, daß die Integer-Einheit (wie alle anderen internen Ressourcen) mit dem PCLOCK (also dem doppelten Takt) arbeiten. Verglichen mit einem MC68020 oder MC68030 sind somit alle Operationen in jedem Fall doppelt so schnell. Durch viele weitere Optimierungen (Ausführung verschiedener Verarbeitungsschritte in Hardware, große Pipeline etc.) wurde erreicht, daß außerdem die Anzahl der Takte, die durchschnittlich für einen Befehl benötigt werden, um den Faktor 1,8 reduziert werden konnte. Dazu existiert in der Integer-Einheit eine sechsstufige Pipeline. Diese wird nicht nur als Prefetch-Queue benutzt, sondern jede Stufe dieser Pipeline kann bereits einen Verarbeitungsschritt ausführen. Diese sind:

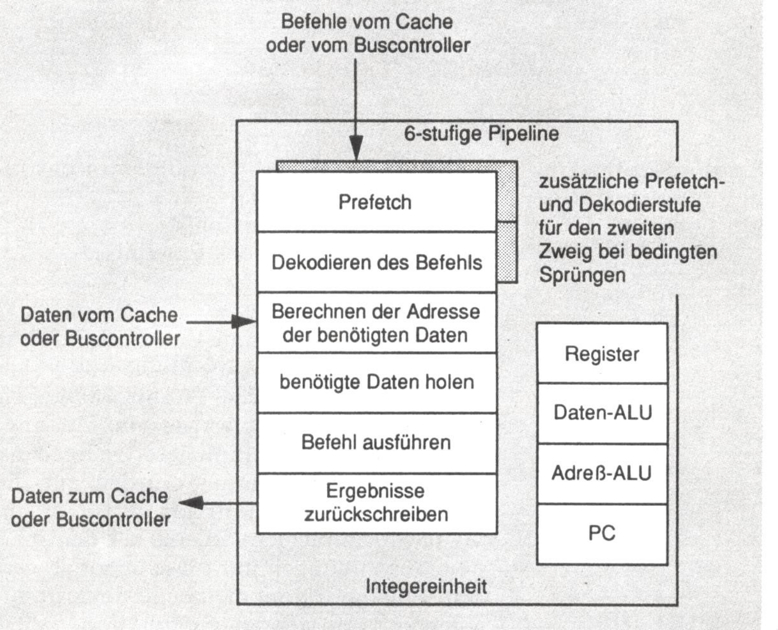
1. Prefetch eines Befehls
2. Befehl dekodieren
3. Die effektive Adresse der benötigten Daten berechnen
4. Die Daten einholen
5. Die Berechnung durchführen
6. Das Ergebnis zurückschreiben

Damit kann diese Pipeline in der Integer-Einheit bereits an bis zu sechs Befehlen gleichzeitig arbeiten.

Eine derartig lange Pipeline hat aber normalerweise auch einen gravierenden Nachteil: Bei jedem ausgeführten Sprung muß sie geleert und wieder neu geladen werden. Da Sprünge sehr häufig sind (alle fünf bis acht Befehle ein Sprung), würde ohne spezielle Vorkehrungen die Leistung deutlich eingeschränkt werden. Diese Nachteile bei Sprungbefehlen vermeidet der MC68040 dadurch, daß die ersten beiden Stufen der Pipeline doppelt vorhanden sind. Wird in der zweiten Stufe (Dekodieren des Befehls) erkannt, daß ein bedingter Sprung vorliegt, so ist zu diesem Zeitpunkt noch nicht bekannt, ob der Sprung ausgeführt wird oder nicht (die Bedingung ist noch nicht ausgewertet). Der MC68040 liest dann beide Programmzweige aus dem Cache und dekodiert bereits die beiden möglichen Befehle nach dem Sprung (den Befehl an der Stelle des Sprungziels, für den Fall, daß der bedingte Sprung ausgeführt werden soll und den Befehl unmittelbar nach dem Sprungbefehl, falls der Sprung nicht ausgeführt wird). Sobald bekannt ist, ob der Sprung ausgeführt wird, ist der erste Befehl aus jedem Zweig bereits geholt und dekodiert. Wird der Sprung ausgeführt, so kann direkt weitergearbeitet werden. Wird der Sprung nicht ausgeführt, braucht nur auf die andere Pipeline umgeschaltet werden, der Befehlsfluß wird dabei nicht unterbrochen. Die Pipeline ist für „ausgeführte bedingte Sprünge“ optimiert, denn Untersuchungen haben ergeben, daß fast 70% aller bedingten Sprünge ausgeführt werden. Das kommt vor allem daher, daß ein bedingter Sprung für das Abbruchkriterium am Ende einer Schleife fast immer ausgeführt wird und nur beim letzten Schleifendurchlauf ein einziges Mal der Sprung nicht ausgeführt wird.

(An)	Adressregister indirekt
(An)+	Adressregister indirekt mit postincrement
-(An)	Adressregister indirekt mit predecrement
(d16,An)	Adressregister indirekt mit 16 Bit Offset
$xxxx	extended Adresse
#xxx	immediate

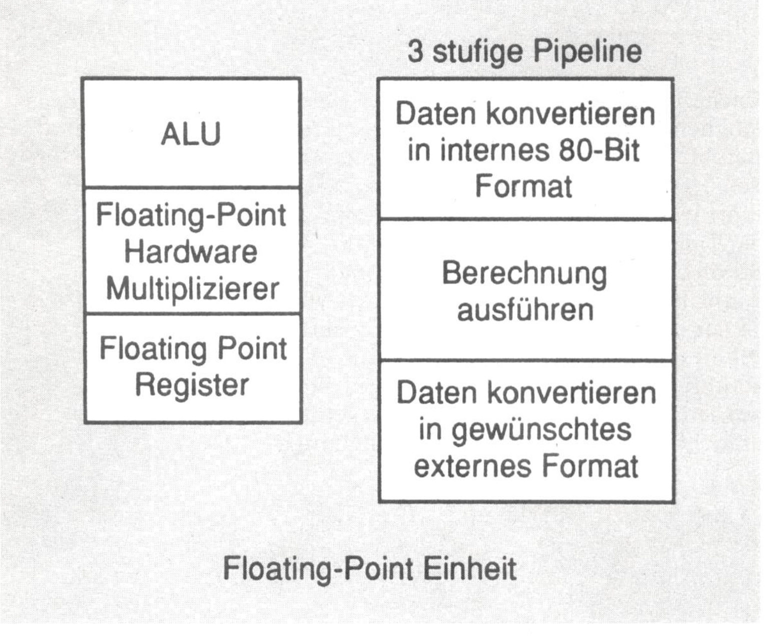
Optimierte Adressierungsarten, die durch einen Hardware-Decoder verarbeitet werden.



Integer-Einheit mit der sechsstufigen Pipeline.

Floating-Point-Einheit

Die Fließkommaeinheit ist voll kompatibel zum IEEE-Standard 754-1985 und arbeitet parallel zur Integer-Einheit. Sie hat eine eigene dreistufige Pipeline und einen eigenen Registersatz mit acht Floating-Point-Registern, die eine Breite von je 80 bit haben und kompatibel zu den Motorola-Prozessoren MC68881/68882 sind. Die drei Pipelinestufen dienen der gleichzeitigen Bearbeitung mehrerer Befehle und haben folgende Aufgaben:



Floating-Point-Einheit mit dreistufiger Pipeline.

1. Konvertieren der Operanden in das interne 80-bit-Format
2. Ausführung des Befehls
3. Zurückschreiben des Ergebnisses

Optimierung

Die Floating-Point-Einheit wurde für die Befehle optimiert, die üblicherweise am meisten benutzt werden. Für maximale Leistung dieser Einheit wurde ein Floating-Point-Hardware-Multiplizierer eingebaut. Für bessere Optimierung wurden nur die Fließkomma-Grundfunktionen im MC68040 implementiert, da bei den meisten Anwendungen nur diese benötigt werden. Für die trigonometrische, logarithmischen und transzendenten Funktionen steht ein Software-Paket zur Verfügung, das vollen Befehlsumfang des MC68881/882 bei voller Kompatibilität gewährleistet. Dieses Paket kann in Form von sogenannten F-Line-Traps implementiert werden. In dem Fall erzeugen alle Floating-Point-Opcodes, die nicht implementiert sind, eine neue Exception (nicht implementierter Floating-Point-Befehl). Diese Exception besitzt ein neues Stackformat. Die effektive Adresse der Operanden wird schon vom MC68040 berechnet, der Eingangsoperand wird bereits in das interne Extended-Precision-Format umgewandelt und auf dem Stack mit abgelegt. Die Software-Emulation kann diese Operanden sofort benutzen (ohne sie aufwendig im Speicher zu lokalisieren und in das benutzte Format umzurechnen) und damit sehr effizient arbeiten. Sie kann aber auch in Form von Libraries benutzt werden, was der Geschwindigkeit zusätzlich zugute kommt, da kein Exception-Overhead entsteht.

Caches

Ein schneller Mikroprozessor ist sehr abhängig von der Geschwindigkeit der Speicher. Jeder Befehl, der ausgeführt werden soll, muß zuerst aus dem Speicher gelesen werden. Genauso müssen Daten, die verarbeitet werden sollen, erst aus dem Speicher gelesen und später wieder zurücktransportiert werden. Um von der Speichergeschwindigkeit nicht mehr als unbedingt nötig abhängig zu sein, besaß bereits der MC68020 ein 256 Byte großes Befehlscache. Der MC68030 ging einen Schritt weiter und vereinigte in sich sowohl ein Befehls-, als auch ein davon unabhängiges Datencache. Beide wurden als logische Caches (sie befinden sich am logischen Adreßbus zwischen CPU und MMU) mit einer Größe von je 256 Byte ausgeführt.

Beim MC68040 dient der größte Teil der 1,2 Millionen Transistoren den Caches und deren Verwaltung. Es gibt wieder getrennte Befehls- und Datencaches mit eigenen Bussen, so daß auf beide Caches gleichzeitig zugegriffen werden kann. Hier wurden physikalische Caches implementiert, sie befinden sich am physikalischen Adreßbus der MMU. Dies bietet verschiedene Vorteile:

Im Multitasking-Betrieb überschneiden sich meist die logischen Adreßbereiche der verschiedenen Tasks, falls eine MMU verwendet wird. Ein logisches Cache kann nicht zwischen Speicherbereichen verschiedener Tasks unterscheiden, wenn sie gleiche (logische) Adressen haben. Das hat zur Folge, daß bei jedem Task-Switch ein logisches Cache von der Software gelöscht werden muß. Bei einem Copy-Back-Cache müßten alle veränderten Daten in den Speicher zurückgeschrieben werden, was bei einem großen Cache viel Zeit kostet. Damit erhöhen sich die für einen Task-Switch notwendigen Zeiten drastisch, ein Echtzeitbetrieb wäre ausgeschlossen.

Keine Nachteile

Ein physikalisches Cache, wie es vom MC68040 benutzt wird, kennt derartige Nachteile nicht. Da sich die physikalischen Adressen verschiedener Tasks immer unterscheiden, können Speicherausschnitte beliebig vieler Tasks gleichzeitig im Cache

gehalten werden. Bei einem Task-Switch muß das Cache also nicht gelöscht werden, und veränderte Daten brauchen nicht zurückgeschrieben werden (falls das Cache im Copy-Back-Modus arbeitet). Außerdem steigt dadurch auch die Trefferwahrscheinlichkeit im Cache.

Aufbau und Betriebsarten

Beide Caches haben eine Größe von je 4 KByte und sind vierfach setassoziativ ausgeführt. Die Caches sind für Burst-Betrieb optimiert, also für die Aufnahme des Datentyps „Cachezeile“ (16 Byte) vorgesehen. Für jede Cachezeile existiert ein „Valid“-Bit, das die Gültigkeit bzw. Ungültigkeit der gesamten Zeile repräsentiert. Der Aufbau einer Zeile des Befehlscaches und einer Zeile des Datencaches ist unterschiedlich.

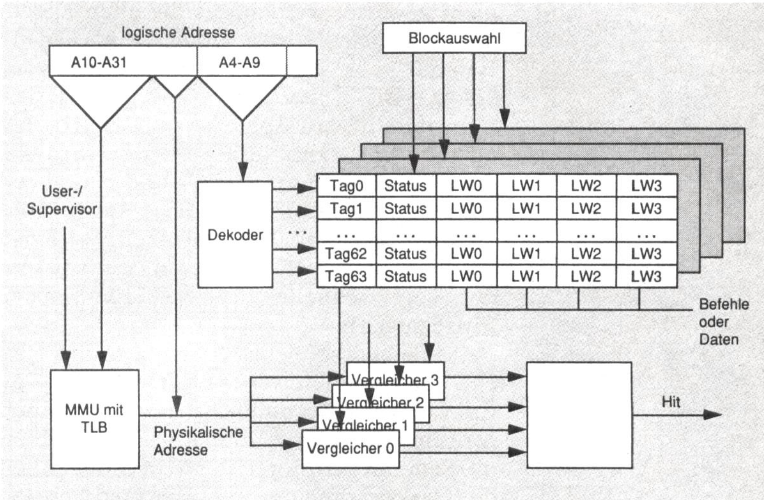
In jeder Datencachezeile existieren zusätzlich vier sog. „Dirty“-Bits, die je einem Langwort innerhalb dieser Zeile zugeordnet sind. Die „Dirty“-Bits zeigen für jedes einzelne Langwort getrennt an, ob sein Wert mit dem entsprechenden Wert aus dem Speicher noch übereinstimmt.

Das Datencache kann sowohl im Write-through- als auch im Copy-Back-Modus arbeiten. Die Arbeitsweise wird für jede Speicherseite beziehungsweise für jedes transparente Fenster getrennt festgelegt (siehe Beschreibung der MMU) und kann damit optimal auf die Anwendung abgestimmt werden.

Write-through-Betrieb

Im Write-through-Betrieb werden Schreibzugriffe auf den Speicher direkt an den Speicher abgesetzt (das kann allerdings durch die Buffer in der Bussteuerung um einige Zyklen verzögert werden). Falls die Adresse des Schreibzugriffs bereits im Cache vorhanden ist, wird der entsprechende Cacheeintrag mit den neuen Daten überschrieben. Ist unter dieser Adresse kein Eintrag vorhanden, so wird durch den Schreibzyklus kein neuer Eintrag erzeugt (No Write Allocate). In diesem Modus ist Datenkonsistenz zwischen Cache und Speicher bei Schreibzugriffen des Prozessors gewährleistet, denn die Daten werden immer sowohl in das Cache (falls ein Eintrag vorhanden war) als auch in den Speicher geschrieben.

Der Copy-Back-Modus dient dazu, die Belastung des externen Busses nochmals zu verringern. Bei jedem Schreibzugriff werden die Daten, falls die gewünschte Adresse bereits vorhanden ist, nur in das Cache abgelegt. Ist die Adresse noch nicht vorhanden, so wird vorher die vollständige Cachezeile aus dem Speicher in das Cache übertragen und anschließend werden die zu schreibenden Daten nur in



Aufbau der Befehls- bzw. Dateneinheit mit MMU und Cache.

das Cache abgelegt. Bei jedem Schreibzugriff auf eine Adresse, die als Copy-Back definiert ist, wird das entsprechende „Dirty“-Bit gesetzt. Damit sind die Daten als geändert gekennzeichnet und stimmen nicht mehr mit den Daten im Speicher überein. Die Daten werden, sobald diese

Tag	V	LW3	LW2	LW1	LW0

Tag: 22 Bit physikalische Adreßmarke
V: Valid-Bit für die Zeile
LWx: 32-Bit Datenwert (Langworte)

Aufbau einer Zeile im Befehlscache. Da der Buscontroller versucht, immer einen Burst-Zugriff zu machen, genügt ein Gültigkeitsbit pro Zeile.

Tag	V	LW3	D3	LW2	D2	LW1	D1	LW0	D0

Tag: 22 Bit physikalische Adreßmarke
V: Valid-Bit für die Zeile
LWx: 32-Bit Datenwert (Langworte)
Dx: Dirty-Bit für Langwort x

Aufbau einer Zeile im Datencache. Neben einem Gültigkeitsbit pro Zeile gibt es für jedes einzelne Langwort ein „Dirty“-Bit.

Cachezeile anderweitig benötigt wird, automatisch in den Speicher zurückgeschrieben. Sie können auch per Software durch den Befehl CPUSH gezielt in den Speicher kopiert werden. Typische Anwendungen für den Copy-Back-Modus sind lokale Datenstrukturen und Stacks.

Daten werden aus dem Speicher immer in Form eines Bursts übertragen (der Buscontroller kann aus einem Burst aber vier einzelne Langwort-Transfers machen). Erfolgt dieses Übertragen der Daten in eine Cachezeile, die bis dahin andere Daten enthielt und von denen mindestens ein Langwort als „Dirty“ gekennzeichnet war, so wird diese Zeile vorher in den eigens dafür vorgesehenen Push-Buffer kopiert und erst nach dem Lesezyklus, sobald der Bus frei ist, in den Speicher transferiert. Damit wird ein Lesezugriff sofort ausgeführt und die Daten stehen sofort zur Verfügung. Es muß

nicht gewartet werden, bis die alten Daten in den Speicher zurückgeschrieben wurden. Ist nur bei einem einzigen Langwort „Dirty"-Bit gesetzt, so wird nur dieses Langwort in den Speicher transferiert. Ist mehr als ein „Dirty"-Bit gesetzt, so wird die ganze Zeile in Form eines Burst zurückgeschrieben. Der Grund für diese Art der Implementierung ist: Ein Einzelzugriff ist schneller als ein Burst, aber ein Burst ist in der Regel schneller als zwei (oder mehr) Einzelzugriffe.

Speicherbereiche, die nicht in das Cache übernommen werden dürfen (etwa Peripherie oder bestimmte Bereiche in einer Multiprozessoranwendung), können über die Seitendeskriptoren der MMU oder über die Deskriptoren der transparenten Fenster als nicht cachefähig gekennzeichnet werden und erzeugen dann keinen Eintrag im entsprechenden Cache. Unabhängig vom ausgewählten Modus werden Read-Modify-Write-Zugriffe nicht in das Cache übernommen. Sollten die angesprochenen Daten bereits im Cache stehen, so werden sie trotzdem nicht aus dem Cache, sondern aus dem Speicher gelesen. Diese Zugriffe werden erzeugt durch die Befehle TAS, CAS und CAS2, sowie bei Tabellendurchläufen der MMU erzeugt.

Datenkonsistenz bei mehreren Busmastern

Existiert ein zweiter Busmaster (zum Beispiel ein DMA-Controller), gibt es in der Regel Probleme mit der Datenkonsistenz zwischen Cache und Speicher. Diese lassen sich entweder durch sehr großen Software-Aufwand oder durch Hardware-Unterstützung lösen. Letzteres ist nur bei physikalischen Caches möglich. Der MC68040 bietet diese Hardware-Unterstützung mit der Snoop-Logik.

Es gibt zwei verschiedene Klassen von Problemen, die bei einem Bus mit mehreren Mastern auftreten können:

1. Überschreibt ein zweiter Busmaster (etwa ein DMA-Baustein) Daten im Hauptspeicher, und der Mikroprozessor hat davon eine Kopie im Cache, so arbeitet der Mikroprozessor in Zukunft mit den – inzwischen nicht mehr aktuellen – Daten aus dem Cache weiter, da nicht mehr auf den – inzwischen veränderten – Speicher zugegriffen wird. Dieses Problem ließe sich dadurch auslösen, daß für alle die Speicherbereiche, die nicht exklusiv für den Prozessor reserviert sind, das Cache abgeschaltet wird, aber im Sinne höchster Leistung ist diese Lösung wenig sinnvoll.
2. Wenn das Cache im Mikroprozessor im Copy-Back-Modus arbeitet, sind die Daten im Cache in der Regel nicht konsistent mit den Daten im Speicher. Würde ein zweiter Busmaster die Daten im Speicher lesen, so sind diese unter Umständen nicht gültig, da sie der Mikroprozessor inzwischen verändert, aber nicht in den Speicher zurückgeschrieben hat. Im Write-through-Modus wäre das Problem zwar gelöst, aber durch die höhere Busbelastung sinkt die Verarbeitungsleistung des gesamten Systems.

Der MC68040 löst beide Probleme mit seiner Snoop-Logik. Diese überwacht ständig den Bus und stellt Abweichungen zwischen Cache- und Speicherinhalt fest und – falls gewünscht – werden diese Abweichungen sofort bereinigt. Die Snoop-Logik wird durch die Snoop-Control-Eingänge (SC0 und SC1) eingeschaltet und gesteuert. Ist sie eingeschaltet und der MC68040 ist nicht Busmaster, so werden alle Bussignale (Adreß-, Datenbus und Steuersignale) als Eingang geschaltet. Wird jetzt von einem anderen Busmaster (etwa einem DMA-Baustein) ein Schreibzugriff auf den Speicher durchgeführt, so wird das von der Snoop-Logik festgestellt. Diese prüft nun, ob die angesprochene Adresse im Cache enthalten ist. Ist sie nicht im Cache, so wird nichts getan; ist sie enthalten, so wird entweder die entsprechende Cachezeile für ungültig erklärt (das V-Bit wird in dieser Zeile gelöscht), oder diese Cachezeile wird mit den Daten, die auf dem Bus sind, aktualisiert, je nachdem welche Arbeitsweise durch die Snoop-Control-Eingänge festgelegt ist. Jede Veränderung des Speichers, die durch einen anderen Busmaster hervorgerufen wird, hat also auch Einfluß auf das Cache, falls die angesprochene Adresse im Cache enthalten war. Die alten Daten werden entweder für ungültig erklärt, oder das Cache wird automatisch auf den neuesten Stand gebracht. Damit ist sichergestellt, daß sich im Cache keine ungültigen Daten befinden.

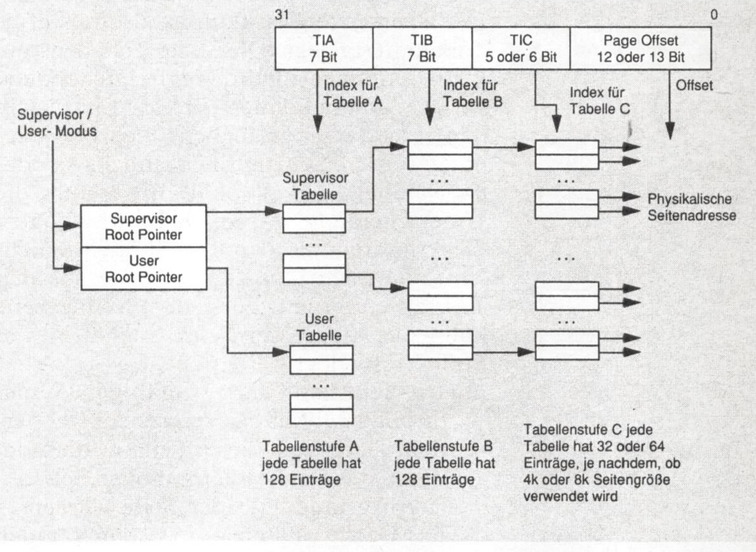
Im zweiten Fall liest ein zweiter Busmaster die Daten aus dem Speicher und überträgt diese beispielsweise an eine Peripherieeinheit. Arbeitet der MC68040 innerhalb dieses Bereiches im Write-through-Modus, so ergibt sich daraus kein Problem. Wird jedoch dieser Bereich im Copy-Back-Modus betrieben, so stimmt der Speicherinhalt nicht unbedingt mit dem Cache überein, denn die Änderungen, die der MC68040 durchführt, ändern nur die Daten im Cache, nicht im Speicher. Führt nun ein zweiter Busmaster einen Lesezyklus auf den Speicher durch, so überwacht die Snoop-Logik auch diesen. Der MC68040 aktiviert nun im voraus die Leitung /MI (Memory Inhibit), solange die Snoop-Logik über die Eingänge SC0 und SC1 eingeschaltet ist und der MC68040 nicht Busmaster ist. Durch /MI wird nun ein Speicher daran gehindert, auf den Lesezyklus des

zweiten Busmasters zu reagieren. Statt dessen prüft der MC68040, ob die gewünschte Adresse im Cache enthalten ist. Ist sie enthalten, werden die zugehörigen Daten vom MC68040 (obwohl er nicht Busmaster ist) auf den Bus gelegt, der Zyklus durch Aktivieren von /TA (Transfer Acknowledge) beendet (der MC68040 verhält sich in diesem Fall wie der Speicher) und der zweite Busmaster arbeitet mit den aktuellen Daten. Ist die gewünschte Adresse nicht im Cache des MC68040 enthalten, so wird die Leitung /MI inaktiv geschaltet, so daß der Speicher reagieren kann.

Memory-Management

Der MC68040 bietet zwei voneinander unabhängige, physikalische Caches, eines für Befehle und eines für Daten. Jedes besitzt einen eigenen Adreß- und Datenbus, damit in jedem Takt ein gleichzeitiger Zugriff auf beide Caches möglich ist. Da hier physikalische Caches implementiert sind, existieren natürlich auch zwei unabhängige MMU, so daß gleichzeitig Adreßumsetzungen für beide Caches erfolgen können. Jede MMU hat einen eigenen Übersetzungs-Cache (TLB = Translation Lookaside Buffer) mit einer Größe von je 64 Einträgen. In den TLB werden die zuletzt benutzten Seitendeskriptoren für die Umsetzung gespeichert. Die Adreßübersetzung erfolgt gleichzeitig

Tabellenstruktur beider MC68040-MMU.



zu den Cachezugriffen in Hardware, so daß nur minimale Zeit dafür benötigt wird. Die Implementierung in Hardware hat zur Folge, daß einige Änderungen im Supervisorbereich gegenüber dem MC68030 gemacht werden mußten (der MC 68030 führt Adreßübersetzungen in Mikrocode durch). Die Übersetzungsstruktur ist als dreistufige Tabelle festgelegt. Die letzte Tabellenstufe kann optional als indirekter Pointer benutzt werden. Damit können einzelne Einträge in den Übersetzungstabellen mehrfach benutzt werden. Seitengrößen sind als 4k oder 8k wählbar. Ein „Global"-Bit erlaubt die Unterscheidung von lokalen und globalen Deskriptoren und damit auch die Möglichkeit, alle nicht globalen Einträge aus den TLB zu entfernen. Für jede einzelne Seite kann die Arbeitsweise des Caches in vier Stufen festgelegt werden:

- Die Seite kann als cachefähig im Writethrough-Modus gekennzeichnet werden. Daten aus dieser Seite werden damit in das Cache übernommen. Schreibzugriffe innerhalb der Seite werden in das Cache und gleichzeitig auch in den Speicher ausgeführt.
- Eine weitere Möglichkeit ist: cachefähig im Copy-Back-Modus. Daten aus dieser Seite werden damit in das Cache übernommen; Schreibzugriffe innerhalb der Seite werden nur in das Cache durchgeführt und benötigen damit den externen Bus nicht. Die Daten können dann auf eine spezielle Anforderung in den Speicher in Form von Burst-Transfers zurückgeschrieben werden. Wird die Cachezeile für andere Werte benötigt, so werden die alten Daten vorher automatisch in den Speicher geschrieben.

31 4 3 2 1 0

Adresse der nächsten Tabelle	x x x x	U	W	UDT

Deskriptor, wie in Stufe A oder B enthalten

31 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Phys. Seitenadresse	UR	G	UPA	W	UPA	M	U	W	UPDT

Page-Deskriptor in Stufe C, für eine Seitengröße von 8k

31 12 11 10 9 8 7 6 5 4 3 2 1 0

Phys. Seitenadresse	UR	G	UPA	W	UPA	M	U	W	PDT

Page-Deskriptor in Stufe C, für eine Seitengröße von 4k

UR:	reserviert für Benutzer	M:	Modified
G:	Global	U:	Used
U1, U0:	User Page Attribute	W:	Write-protect
S:	nur Supervisor	UDT:	Upper Descriptor Type
CM:	Cache-Modus	PDT:	Page Descriptor Type

UDT und PDT zeigen an, ob die nächste Tabelle bzw. die Page resident im Speicher stehen und ob der Page-Descriptor als indirekter Pointer benutzt werden soll

Aufbau der verschiedenen Deskriptoren der MMU.

- Nicht cachefähig, „serialized". Die Daten werden nicht in das Cache übernommen. Der Bus des MC68040 ist so ausgelegt, daß Lesezugriffe stets Priorität haben, da die Daten, die vom Bus gelesen werden, unmittelbar benötigt werden. Das bedeutet, daß Schreibzugriffe verzögert werden können. In der Bussteuerung des MC68040 befinden sich deshalb Writebuffer, die bis zu drei unabhängige Schreibzugriffe zwischenspeichern können. Damit entspricht aber die Reihenfolge der Lese- und Schreiboperationen nicht der Anordnung im Programm. Diese interne Umordnung der Zugriffe kann mit dem Modus „serialized" abgeschaltet werden. Dies ist für Speicherseiten von Peripheriebausteinen notwendig.
- Nicht cachefähig, „not serialized". Die Daten werden nicht in das Cache übernommen. Die Reihenfolge von Lese- und Schreiboperationen wird optimiert.

Jeder Deskriptor stellt außerdem zwei Bits zur Verfügung, die vom Anwender beliebig benutzt werden können. Bei jedem Zugriff auf eine Adresse innerhalb einer Speicherseite erscheinen diese beiden Bits an den dafür vorgesehenen Pins des MC68040.

Bus-Interface

Der Buscontroller ist eine eigenständige Einheit des MC68040. Er kann parallel zu den übrigen Einheiten arbeiten und stellt das Interface zwischen dem externen Bus und den internen Einheiten (Integer- und Floating-Point-Unit, MMU und Caches) dar. Er aktiviert nach außen hin alle Bussignale, überwacht die Datenübertragung und stellt das Bindeglied zwischen dem internen (PCLOCK) und dem externen Takt (BCLOCK) dar. Er kann nach außen zwei verschiedene Arten von Zugriffen durchführen: Einerseits Einzelzugriff auf ein Byte, ein Wort oder ein Langwort. Dieser Zugriff betrifft immer nur eine einzelne Speicherzelle. Der Speicher muß immer 32 bit breit sein, eine dynamische Busbreitensteuerung, wie sie der MC68020/68030 bietet, ist nicht implementiert. Die zweite Art von Zugriff ist ein Burst-Transfer. Hier werden 16 Byte (also vier Langworte) am Stück übertragen. „Misaligned" Datentransfers werden vom Buscontroller in mehrere einzelne Byte- und Worttransfers aufgelöst und als Einzeltransfers durchgeführt. Im Hinblick auf maximalen Datendurchsatz sollen aber „misaligned" Daten sowenig wie möglich benutzt werden, da dabei immer mehr als ein Zugriff erforderlich ist.

Der Buscontroller ist dafür ausgelegt, die Verarbeitungseinheiten so zu bedienen, daß diese ihre maximale Leistung entfalten können. Dazu gibt er allen angeforderten

Lesezugriffen Priorität gegenüber anstehenden Schreibzugriffen. Um das zu ermöglichen, besitzt er eine dreistufige Schreib-Pipeline. Eine Stufe davon ist in der Lage, eine Cachezeile (also 16 Byte) aufzunehmen und speichert sowohl die Daten der gesamten Cachezeile als auch ihre Adresse und Zugriffsattribute wie beispielsweise Transfer Modifier, sowie vom Anwender definierte Page-Attribute. Die beiden anderen Stufen können maximal ein Langwort mit Adresse und Attributen aufnehmen.

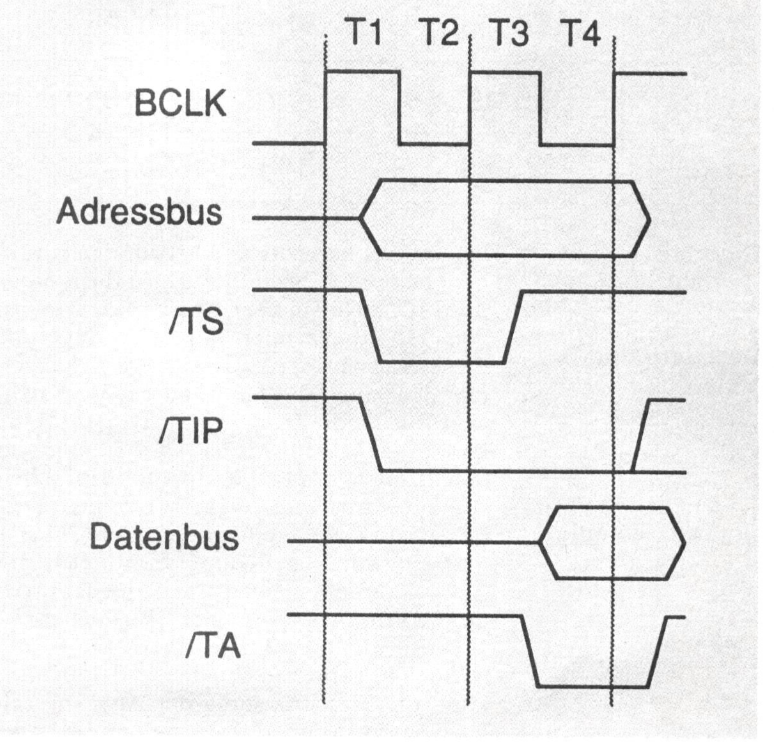
Der Buscontroller

Ein von den Verarbeitungseinheiten oder dem Datencache angeforderter Schreibzyklus legt die Daten nur in diesen Schreibpuffer ab, danach sind die internen Busse wieder frei. Der Buscontroller schreibt dann die Daten selbständig in den Speicher, sobald der Bus frei ist, oder ein Schreibzugriff ansteht und keine Stufe im Puffer mehr frei ist. Durch diese Pipeline erscheinen die Datenzugriffe auf dem externen Bus nicht in ihrer „natürlichen", vom Programm festgelegten Reihenfolge. Das bringt zwar maximale Ausnutzung des Busses, muß aber für ganz bestimmte Bereiche (beispielsweise für die Peripherie) abgeschaltet werden können. Hier muß sichergestellt sein, daß zum Beispiel ein Statusregister erst gelesen wird, nachdem ein Kontrollregister beschrieben wurde. Dazu gibt es in jedem Seitendeskriptor die Möglichkeit, für diese Speicherseite Zugriffe in der „natürlichen" Reihenfolge zu erzwingen. Wird dieses „serialized" Bit gesetzt, erscheinen alle Zugriffe innerhalb dieser Seite in der gleichen Reihenfolge, wie sie von der Software angefordert wurden.

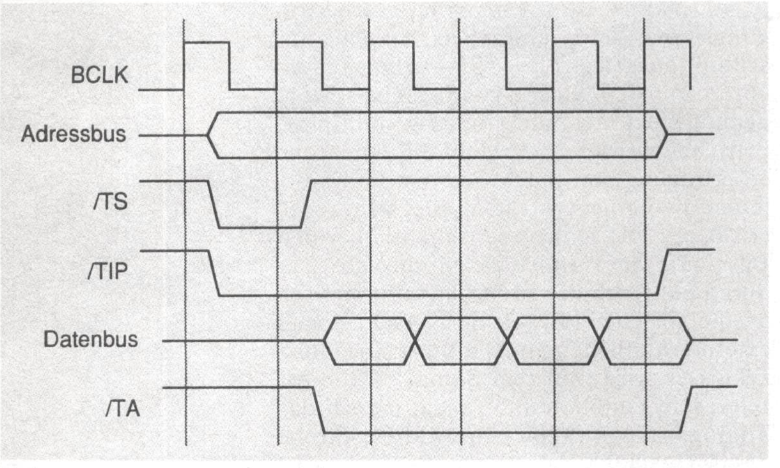
Der Buscontroller ist auch in der Lage, Burst-Zugriffe in vier Einzelzugriffe umzusetzen, falls er von der angesprochenen externen Einheit durch die Leitung „/TBI" (Transfer Burst Inhibit) dazu aufgefordert wird. Dieser „aufgebrochene" Burst wird beim MC68040 „Fake Burst" genannt. Werden von einer Verarbeitungseinheit Daten angefordert und nicht im Cache gefunden, versucht der Buscontroller einen Burst-Zugriff auf den Speicher, um den Bus optimal zu nutzen.

Die Daten an der gewünschten Adresse werden dabei als erstes durch den Burst oder den „Fake Burst" übertragen und sofort an die Einheit weitergeleitet, die die Daten angefordert hat, ohne daß das Ende des Burst-Zugriffs abgewartet werden muß. Anschließend werden die restlichen Daten zyklisch innerhalb des 16 Byte großen Burst-Blocks übertragen. Die erste Adresse, die bei einem Burst übertragen wird, ist deshalb nicht die erste Adresse innerhalb des übertragenen Burst-Blocks.

Der Bus ist für maximale Übertragungsraten und einfachen Speicheranschluß ausgelegt. Der Bus arbeitet synchron mit dem Bustakt und kann sowohl gemultiplext als auch nicht-gemultiplext betrieben werden. Er erlaubt die synchrone Übertragung von Bytes, Worten, Langworten und vom neuen Datentyp „Cachezeile" (16 Byte). Die Datenübertragung über den Bus läuft im Handshake-Modus ab. Dadurch paßt sich der MC68040 automatisch an die Geschwindigkeit der verschiedenen externen Einheiten an. Der Start eines Buszyklusses wird mit dem Signal /TS angezeigt. Die Funktionalität, nicht jedoch das Timing, ist in etwa mit dem Address-Strobe des MC68000 vergleichbar. Beendet wird der Buszyklus mit dem Signal /TA (Transfer Acknowledge), das vom Speicher bzw. von der Peripherie zurückgemeldet wird. /TA weist gewisse Ähnlichkeiten mit dem /DTACK des MC68000 auf, muß jedoch synchron zum Takt aktiviert werden. Außerdem wird /TA mit der gleichen Taktflanke in den MC68040 übernommen, wie die Daten. Da /TA (wie fast alle anderen Eingangssignale) synchron zum Takt aktiviert werden muß, wird keine Zeit für interne Synchronisierung benötigt. Die Ausgangssignale werden, abgesehen von wenigen Ausnahmen, mit der steigenden Flanke des Bustaktes gesetzt. Die Eingangssignale (außer den asynchronen Signalen IPLx und RESIN) werden mit der steigenden Flanke übernommen. Ein einzelner Buszugriff benötigt zwei Takte; die möglichen vier Zustände werden mit T1, T2, T3 und T4 be-



Schematischer Ablauf eines Einzelbuszugriffs. Mit der steigenden Flanke von T4 werden sowohl das TA-Signal als auch die Daten übernommen.



Schematischer Ablauf eines Burst-Zugriffs.

zeichnet. Der schematische Ablauf eines Einzelbuszugriffes ist in Bild 8a, eines Burst-Zugriffs in Bild 8b dargestellt. Die Ausgangssignale, zum Beispiel bei einem Langwort-Lesezyklus, werden kurz nach steigender Flanke von T1 stabil. Die Daten werden dann mit der steigenden Flanke von T4 vom Prozessor übernommen.

Die Busgeschwindigkeit hängt neben der Zugriffszeit der Speicher vor allem auch von den internen und externen Verzögerungszeiten der einzelnen Leitungen ab. Diese Verzögerungszeiten werden vor allem durch die Treiberleistung der Ausgangszellen und der kapazitiven Last auf den einzelnen Leitungen beeinflußt. Die Bustimings des MC68040 sind für eine kapazitive Last von 130 pF definiert. Eine Verringerung dieser Last bringt zwar kürzere Schaltzeiten, aber auch die Notwendigkeit zusätzlicher Buffer, die wiederum Zeit kosten. Eine andere Möglichkeit der Verkürzung der Schaltzeiten besteht darin, die Treiberleistung der Ausgangszellen zu erhöhen. Dies bringt aber Nachteile wie größere Über-/Unterschwingungen und Störungen mit sich, außerdem steigen die Anforderungen an die Stromversorgung des Bausteins.

	Einzel- oder erster Zugriff		
Waits	Low-Drive Modus	High-Drive Modus	Burst
0	45 ns	54 ns	35 ns
1	85 ns	94 ns	35 ns
2	125 ns	134 ns	35 ns

Theoretische Speicherzugriffszeiten (ohne Berücksichtigung von Buffern, Decodern und Steuerlogik), die bei einem MC68040 im Low- bzw. im High-Drive-Modus für verschiedene Waitstates notwendig sind.

Für die optimale Anpassung des MC68040 an verschiedene Bedürfnisse kann die Treiberleistung der Ausgangszellen vom Anwender eingestellt werden. Es existiert ein sogenanntes High-Drive- und ein Low-Drive-Modus, der getrennt für Adreß-, Daten- und Controlbus einstellbar ist. Die Einstellung erfolgt während des RESETS durch den Zustand an den IPL-Eingängen. Damit könnte etwa der Adreß- und Controlbus im High-Drive-Modus und der Datenbus im Low-Drive-Modus benutzt werden.

Im Low-Drive-Modus sind die Schaltzeiten langsamer, aber es vereinfacht sich die Stromversorgung und in der Regel ist keine aufwendige Busterminierung erforderlich. Der High-Drive-Modus ist zwar schneller, verlangt aber wesentlich aufwendigere Stromversorgung und präzise Busterminierung.

Busarbitrierung

Die Busarbitrierung arbeitet in einer anderen Art und Weise als beim MC68020/030. Ein MC68030 beispielsweise ist durch Voreinstellung Eigentümer des Busses. Wird der Bus von einer anderen Einheit benötigt, so zeigt diese das durch Aktivierung der Bus-Request-Leitung an. Der MC68040 ist nicht per Definition Busmaster. Benötigt er den Bus, so muß er diesen durch die Leitung Bus-Request (hier ein Ausgang!) anfordern. Ein Arbiter teilt den Bus (durch Aktivierung von Bus Grant) zu und der MC 68040 behält ihn solange, bis er vom Arbiter wieder zurückgefordert wird (Inaktivierung von Bus Grant).

Statussignale für Debug-Unterstützung

Je leistungsfähiger ein Prozessor ist, desto schwieriger wird es, den Programmablauf zu verfolgen. Das kommt zum einen von der hohen Busfrequenz, viel mehr Einfluß hat aber die Architektur. Eines der Hauptziele bei einem modernen Mikroprozessor besteht darin, ihn soweit wie irgend möglich vom Bus unabhängig zu machen. Hier stellt bereits die dreistufige Pipeline eines „normalen" MC68000 Probleme dar. Die Befehle, die vom Bus gelesen werden, werden (wenn überhaupt) erst später ausgeführt, Rückschlüsse von den gerade ablaufenden Buszyklen auf die Befehle, die im Augenblick ausgeführt werden, sind also schwierig. Kommt ein Cache hinzu (wie etwa bei einem MC68020), geben die Bussignale noch weniger Auskunft über die gerade ausgeführten Befehle, da es nach außen hin nicht sichtbar ist, welche Zugriffe auf das Cache zu welchen Zeitpunkten erfolgen. Es ist (vom Gesichtspunkt der maximalen Leistungsfähigkeit aus betrachtet) nicht einmal sinnvoll, diese Zugriffe auf das Cache sichtbar zu machen, denn dadurch wäre der Bus für das „Sichtbarmachen des Zugriffs" belegt. Gewünscht wird aber maximale Parallelarbeit, das heißt

wenn ein Zugriff auf das Cache abläuft, soll gleichzeitig ein Buszugriff auf den Speicher möglich sein.

Ein MC68040 hat nicht nur eine, sondern mehrere Pipelines, er hat nicht nur ein, sondern zwei Caches, zwei MMU und eine Schreib-Pipeline im Businterface. Das führt dazu, daß:

- viele Speicherzugriffe vom Cache aus bedient werden und auf dem Bus nicht erscheinen
- die sichtbaren Zyklen erscheinen verzögert und in einer optimierten Reihenfolge (durch die Schreib-Pipeline).

Diese interne Parallelarbeit macht es so gut wie ausgeschlossen, den Programmablauf anhand der Bussignale zu verfolgen (zum Beispiel mit einem Logikanalysator).

Andererseits ist das Verfolgen des Programmablaufs für Testzwecke dringend erforderlich. Deshalb besitzt der MC68040 vier Statuspins, um den Prozessorstatus und damit den internen Ablauf sichtbar zu machen. Diese Pins heißen PST0, PST1, PST2 und PST3.

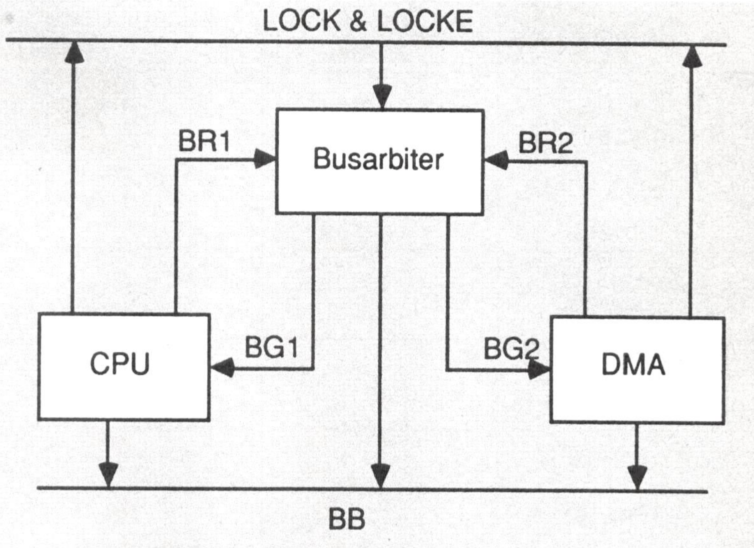
Diese Statuspins zeigen an, ob der Prozessor im User- oder Supervisorbereich arbeitet (an den Transfer-Modifier-Pins ist nur zu sehen, ob der Buszugriff ein User- oder Supervisorzugriff ist), ob ein Befehl beendet oder ein neuer begonnen wurde, ob ein bedingter Sprung aufgetreten ist und ausgeführt worden ist, oder nicht und andere Dinge mehr. Die vollständige Bedeutung der Statuspins ist in der untenstehenden Tabelle aufgelistet.

Taktversorgung

Damit ein Mikroprozessor höchste Leistung bringen kann, muß er einen sehr schnellen Takt benutzen. Mit dem MC 68030 sind bereits seit einiger Zeit Frequenzen bis zu 50 MHz möglich. Andererseits machen es diese hohen Frequenzen immer schwieriger, die Prozessoren in der Praxis anzuwenden. Es muß zum einen nach Hochfrequenzregeln entwickelt werden, zum anderen schlagen die unvermeidlichen Laufzeitverzögerungen durch Logik und Leitungen kräftig zu Buche. Daneben ist es heute nicht möglich, wirtschaftliche Speichersysteme (bestehend aus dynamischen Speichern) mit derartigen Taktraten zu betreiben.

Lösung

Der MC68040 löst dieses Dilemma auf sehr elegante Art und Weise: Das Design wurde so ausgelegt, daß die Verarbeitungseinheiten und der Bus nicht nur durch die Caches, sondern auch vom Takt her entkoppelt wurden. Es gibt zwei Takte: Alle internen Einheiten mit Ausnahme des Busses arbeiten mit dem Prozessortakt (PCLOCK). Der Bus hat einen eigenen Bustakt, den BCLOCK. Beide müssen aber in einem festen Verhältnis zueinander stehen. Der PCLOCK muß exakt doppelt so hoch sein wie BCLOCK. In der Basisversion des MC68040 beträgt der PCLOCK 50 MHz; der Bustakt entsprechend 25 MHz. Diese Verarbeitungsfrequenz von 50 MHz ist bei Motorola keine Neuheit, der MC68030 ist mit dieser Frequenz schon seit einiger Zeit erhältlich. Da für den Designer nur das Zeitverhalten am Bus wichtig ist, wird der BCLOCK als Bezeichnungsmaßstab für die Taktfrequenz des MC68040 benutzt. Ein 25-MHz-MC68040 (das ist die anfängliche Frequenz) benötigt also außer dem 25-MHz-BCLOCK einen 50-MHz-PCLOCK. Damit ist, verglichen mit einem 25-MHz-MC68020 oder -MC68030, bereits ein Faktor zwei an Leistungszuwachs garantiert. Aus allen anderen Optimierungen (mehrere, unabhängige Pipelines, Optimierung der Adressierungsarten, breitere Busse, größere Caches und Abarbeitung einzelner Schritte in Hardware anstelle des Mikrocodes) ergibt sich nochmals ein Faktor von etwa 1,8.



Busarbitrierung des MC68040. Ein externer Arbiter teilt den Bus zu. Der MC68040 muß den Bus genauso über Bus-Request anfordern, wie jede andere Einheit.

Bedeutung der Signale an den Prozessor-Statuspins des MC68040.

PST3	PST2	PST1	PST0	Bedeutung	Modus
0	0	0	0	Befehl gestartet und in Arbeit	User
0	0	0	1	laufender Befehl beendet	User
0	0	1	0	bedingter Sprung nicht ausgeführt	User
0	0	1	1	bedingter Sprung ausgeführt	User
0	1	0	0	MMU-Tabellenzugriff	User
0	1	0	1	Halt (Doppelter Busfehler)	beide
0	1	1	0	reserviert	beide
0	1	1	1	reserviert	beide
1	0	0	0	Befehl gestartet und in Arbeit	Supervisor
1	0	0	1	laufenden Befehl beendet	Supervisor
1	0	1	0	bedingter Sprung nicht ausgeführt	Supervisor
1	0	1	1	bedingter Sprung ausgeführt	Supervisor
1	1	0	0	MMU-Tabellenzugriff	Supervisor
1	1	0	1	Stop-Befehl ausgeführt	Supervisor
1	1	1	0	RTE-Befehl ausgeführt	Supervisor
1	1	1	1	Stackablage bei einer Eception	Supervisor

Befehl	Anzahl Takte MC68040	Anzahl Takte MC68020 / 68881
ADD, SUB, CMP (Register)	1	2
ADD, SUB, CMP (Speicher)	1	6
Branch (ausgeführt)	2	6
Branch (nicht ausgeführt)	3	4
FADD / FSUB	3	51
FADD.S	3	72
FMUL	5	71
FMUL.D	5	98
FDIV	37.5	103

Ausführungszeiten verschiedener Befehle des MC68040 im Vergleich zum MC68030/ MC68882, gemessen in Bustakten.

Boundary-Scan-Test

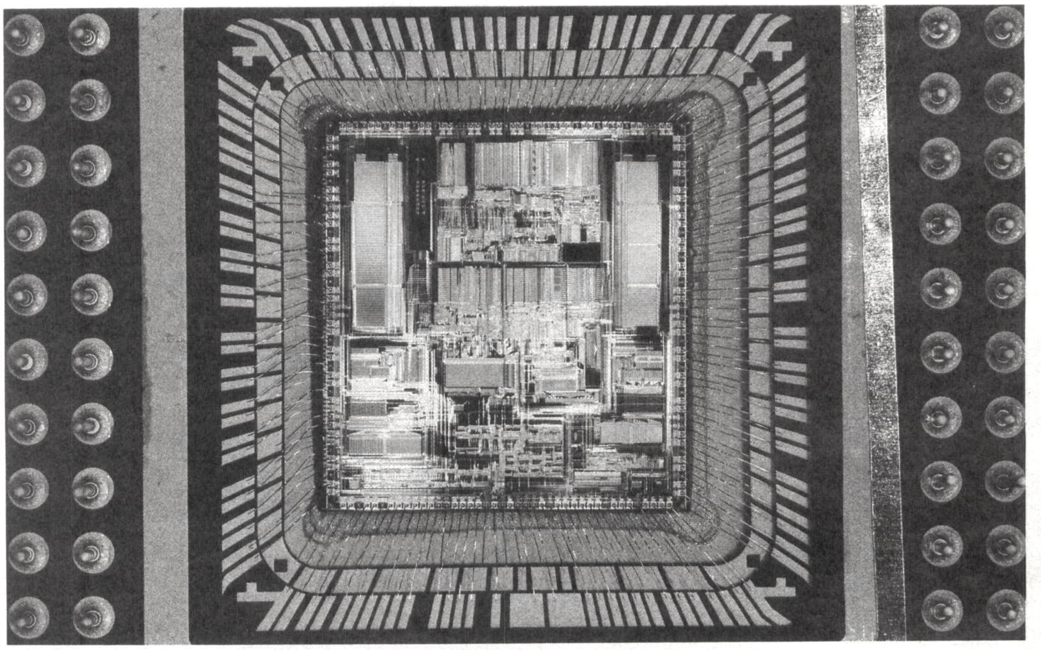
Durch die immer steigende Integration wird die Packungsdichte auf einem Board immer größer. Dadurch wird es auch immer schwieriger, eine Baugruppe zu testen. Visuelle Überprüfung scheidet bei mehrlagigen Boards aus, elektrische Messung von Durchgangswiderständen ist durch die hohe Anzahl der Leitungen und der Pins und daraus folgenden hohen Anzahl von Meßpunkten in der Praxis sehr zeit- und kostenaufwendig. Hinzu kommt, daß oftmals durch beiderseitige SMD-Bestückung der Boards ein Zugang zu den Meßpunkten nicht mehr möglich ist.

Ein für diese Testzwecke standardisiertes Verfahren ist der IEEE-P1149.1-Boundary-Scan-Test, der von der JTAG (Joint Test Access Group) erarbeitet wurde. Dieses Verfahren benutzt die auf dem Board befindlichen Bauteile für den Verbindungstest. Ein Baustein besitzt dazu fünf dedizierte Pins, die es ermöglichen, die Testdaten seriell einzugeben und wieder auszulesen. Der Takt dafür wird über einen eigenen Pin eingespeist und ist völlig unabhängig vom Systemtakt.

Der vereinfachte Ablauf einer derartigen Boundary-Scan Testprozedur sieht wie folgt aus:

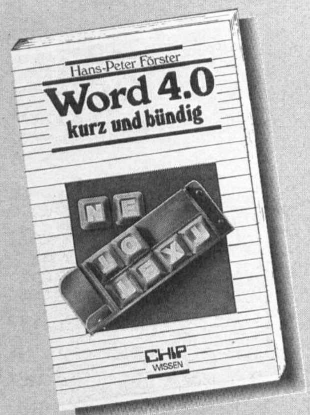
1. Der Baustein wird in den Testmodus geschaltet. Damit werden die Pins von der internen Logik abgekoppelt.
2. Über den seriellen Test-Eingangspin werden mit dem dazugehörenden Takt (der über den Taktpin eingespeist wird) die Daten eingegeben, die an den Pins erscheinen sollen.
3. Die eingegebenen Daten erscheinen an den Pins des angesprochenen Bausteins und werden durch die Verdrahtung auf dem Board an Eingangspins eines anderen (oder auch des gleichen) Chips sichtbar.
4. Diese jetzt an den Eingangspins anstehenden Daten können über den Test-Out-Pin seriell ausgegeben werden.
5. Die so gewonnenen Daten brauchen jetzt nur noch mit den Daten verglichen werden, die aufgrund der Schaltung und der vorher eingegebenen Daten erscheinen sollten. Sind die Daten gleich, ist das Board in Ordnung, ansonsten ist bekannt, welcher Weg fehlerhaft ist.

Der 68040-Mikroprozessor



CHIP WISSEN

Die kompetente Buchreihe rund um den PC



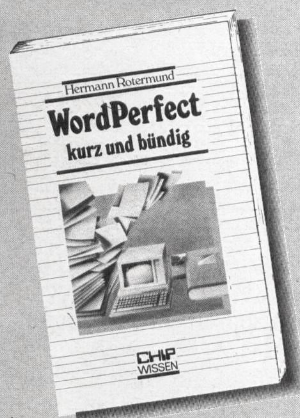
Hans-Peter Förster/Martin Zwernemann:

Word 4.0 kurz und bündig

256 Seiten, 67 Bilder, Hardcover
43,– DM/ISBN 3-8023-**0215**-X

Dieses Arbeitsbuch richtet sich an interessierte Einsteiger sowie an Word-4.0-Anwender, die schnell den Umgang mit dem komfortablen Textverarbeitungsprogramm erlernen wollen. Es ist so aufbereitet, daß es auch als Begleitmaterial für Word-4.0-Schulungen geeignet ist. Aber auch zum Selbststudium ist es in Ergänzung zum Word-4.0-Handbuch ein Hilfsmittel, das viele Tips und Tricks verrät.

- Wichtige Grundfunktionen
- Texte erstellen, korrigieren, speichern
- Texte laden, korrigieren, gestalten
- Texte direkt/indirekt formatieren
- Ausdruck
- Textbausteine
- Rechtschreibprogramm u.a.



Hermann Rotermund:

WordPerfect kurz und bündig

144 Seiten, zahlreiche Bilder, Hardcover
30,– DM/ISBN 3-8023-**0121**-8

In diesem Buch werden zunächst die grundlegenden Arbeitsabläufe vorgestellt, dann die jeweils notwendigen Arbeitsschritte für das Erreichen eines bestimmten Ziels erklärt. Außerdem enthält es Anleitungen zur professionellen Nutzung von WordPerfect für spezielle Aufgaben. Die Erläuterung der Funktionen folgt dem logischen Aufbau des Programms. Dem Buch liegt die deutsche Version 4.2 zugrunde.

Besonderes aus dem Inhalt:

- Makros
- Mischfunktionen
- Rechnen im Text
- Fußnoten und Endnoten
- Inhaltsverzeichnis erstellen, Index und Listen
- Spaltenverarbeitung mit WordPerfect

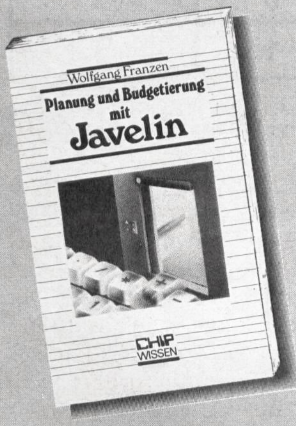


Rudolf Wolff:

Professionell arbeiten mit Ventura Publisher

264 Seiten, zahlr. Bilder, Hardcover
43,– DM/ISBN 3-8023-**0213**-3

Professionell arbeiten mit Ventura Publisher ist sowohl eine Einführung in die Programmvielfalt des Xerox Ventura Publisher, als auch und gerade ein Praxishandbuch, das dem Leser immer dann hilfreich zur Seite steht, wenn Gestaltungs- und Typografieprobleme auftreten. Nach der Einleitung über Voraussetzungen und Möglichkeiten des Ventura-Publisher und einigen Ausführungen zu Satz-, Layout- und Typografiegrundsätzen folgt die ausführliche Einweisung in das Programm, um die angebotenen Möglichkeiten sicher und rationell zu nutzen. Der Praxisteil soll mit Übungen und Beispielen die bis dahin erworbenen theoretischen Kenntnisse vertiefen und dem Anwender Sicherheit vermitteln.



Wolfgang Franzen:

Planung und Budgetierung mit Javelin

280 Seiten, 141 Bilder, Hardcover
48,– DM/ISBN 3-8023-**0206**-0

Den Besitzern von Javelin hilft dieses Buch bei der praktischen Anwendung. Es wurde für alle geschrieben, die ein erprobtes Modell für die Erfolgs- und Liquiditätsplanung und praktische Tips für den erfolgreichen Einsatz von Javelin suchen. Eine Vielzahl von Bildern verdeutlichen die Funktionsweise des Programmpakets. Der Leser kann so jederzeit einzelne Schritte auf dem eigenen PC nachvollziehen.
Eine Diskette mit den im Buch vorgestellten Modellen ist erhältlich.

Haben Sie schon den neuen „CH!P- Katalog“ ?
Bestellen Sie gleich!

VOGEL Buchverlag Würzburg
Postfach 67 40 · 8700 Würzburg 1

Die I/O-Signale

Josef Fuchs

Der Autor
Dipl.-Informatiker Josef Fuchs stammt aus Bogen/Niederbayern. Er studierte an der Technischen Universität München Informatik, Fachrichtung Elektrotechnik, wobei er sich auf Mikroprozessoren spezialisierte. Seit 1986 ist er bei Motorola in München als Field Applications Engineer für die 32-bit-Familie tätig.

Nachfolgend seien noch einige Einzelheiten zum MC68040 geschildert. Es handelt sich um die Bussignale, die ohne Angabe der Polarität näher erläutert werden.

A0-A31: Der Adreßbus ist bidirektional ausgeführt und threestate-fähig. Ist der MC68040 Busmaster, so wird die erste Adresse eines Bustransfers ausgegeben. Ist eine andere Einheit der Busmaster, so wird der Adreßbus als Eingang für die Snoop-Logik benutzt.
D0 – D31: Der Datenbus kann Einzelzugriffe als Byte, Wort oder Langwort übertragen. Während eines Burstzugriffes werden zeitlich hintereinander 128 Bits gelesen beziehungsweise geschrieben.

Transferattribute

R/W (Read/Write): Dieses Signal ist, genau wie die Adressen, bidirektional und wird von der Snoop-Logik überwacht, wenn der MC68040 nicht Busmaster ist.
SIZ0, SIZ1 (Übertragungsgröße): Damit wird die zu übertragende Operandengröße angezeigt. Der angezeigte Wert bezieht sich (anders als beim MC68020/68030) nur auf den aktuellen Zugriff. Auch die Size-Signale werden von der Snoop-Logik überwacht. Ihre Bedeutung zeigt die nachfolgende Tabelle:

SIZ1	SIZ0	Datengröße
0	0	Langwort (4 Byte)
0	1	Byte
1	0	Wort (2 Byte)
1	1	Cache-Zeile (16 Byte)

Signalbedeutung

TT0, TT1 (Transfertyp): Sie werden dazu benutzt, den Typ des augenblicklichen Zugriffs und die Bedeutung der Transfer Modifier Signale anzuzeigen. Sie werden als Eingang zur Steuerung der Snoop-Logik benutzt. Die Snoop-Logik verarbeitet nur normale und MOVE16-Zugriffe. Die Tabelle zeigt die Bedeutung der Signale:
TM0 – TM2 (Transfer Modifier): Damit

TT1	TT0	Transfer Typ
0	0	Normaler Zugriff
0	1	MOVE16 Befehl
1	0	alternate log. Function Codes
1	1	Interrupt / Breakpoint Acknowledge

wird ein Zugriff genauer spezifiziert. Sie sind bei normalen Zugriffen (siehe Transfertyp) vergleichbar mit den Function Codes des MC68000. Bei Interrupt-Acknowledge-Zyklen geben die Transfer Modifier-Leitungen die Ebene des in Arbeit befindlichen Interrupts an. Bei normalen und bei Alternate-Zugriffen haben sie folgende Bedeutung:

TM2	TM1	TM0	Transfer Modifier (Normaler Zugriff)	Transfer Modifier (alt. log. Function Codes)
0	0	0	Daten Cache Push	Log. Function Code 0
0	0	1	User Daten (incl. MOVE16)	Reserviert
0	1	0	User Code	Reserviert
0	1	1	Daten-MMU Tabellenzugriff	Log. Function Code 3
1	0	0	Code-MMU Tabellenzugriff	Log. Function Code 4
1	0	1	Supervisor Daten (incl. MOVE16)	Reserviert
1	1	0	Supervisor Code	Reserviert
1	1	1	Reserviert	Log. Function Code 7

UPA0, UPA1 (User Attribute): Hier stehen bei jedem normalen Zugriff die beiden vom Anwender frei benutzbaren Bits in den Seitendeskriptoren der MMU zur Verfügung.
TLN0, TLN1 (Transfer Line Number): Bei normalen, cachefähigen Zugriffen wird hier angezeigt, in welchen der vier Blöcke der vierfach setassoziativen Caches die Daten abgelegt werden bzw. aus welchem Block sie stammen. Damit ist es möglich, die Caches extern zu duplizieren und damit zusätzliche Snoop-Möglichkeiten aufzubauen.

TLN1	TLN0	Cache-Block
0	0	0
0	1	1
1	0	2
1	1	3

LOCK (Busverriegelung): Das Signal ist vergleichbar mit dem RMC beim MC68020/030. Es zeigt an, daß gerade ein Ready-Modify-Write-Zugriff (TAS, CAS, CAS2 oder ein MMU-Tabellendurchlauf) durchgeführt wird. Diese Leitung muß vom Busarbiter ausgewertet werden. Der MC68040 ist in der Lage, auch während eines Read-Modify-Write-Zugriffs den Bus abzugeben, falls er dazu aufgefordert wird.
LOCKE (Ende Busverriegelung): Hier wird angezeigt, daß der augenblickliche Zugriff der letzte innerhalb einer nicht unterbrechbaren Sequenz ist. LOCKE dient dazu, zwei Read-Modify-Write-Zyklen zu

unterscheiden, da, falls sie unmittelbar aufeinander folgen, LOCK dazwischen nicht negiert wird.
CIOUT (Cache Inhibit Out): Es zeigt das CM-Bit im augenblicklich benutzten Seitendeskriptor beziehungsweise des benützten Fensters an. Es dient zur Steuerung eines externen Caches.

Bussteuerung

TS (Transfer Start): Diese Leitung wird für einen Takt aktiviert, um den Start eines externen Buszyklusses anzuzeigen. Ist der MC68040 nicht Busmaster, so wird TS als Eingang benutzt und von der Snoop-Logik überwacht, um den Start eines externen Buszyklusses festzustellen.
TIP (Transfer in Progress): Hiermit wird angezeigt, daß auf dem Bus ein Datentransfer stattfindet. Es wird dann inaktiv, wenn der MC68040 zwar im Besitz des Busses ist, aber zu diesem Zeitpunkt keinen Zugriff durchführt. Zwischen zwei aufeinanderfolgenden Buszyklen wird TIP nicht inaktiv.
TA (Transfer Acknowledge): TA dient als Handshake-Signal zum TS. Es wird von der angesprochenen Einheit (synchron zum Bustakt) gesetzt und legt das Ende eines Buszyklusses fest. Ist der MC68040 nicht Busmaster, wird TA von der Snoop-Logik als Eingang benutzt, um das Ende eines Buszyklusses festzustellen. TA kann auch von der Snoop-Logik als Ausgang benutzt werden, wenn der MC68040 aufgrund von „dirty" Daten den Speicher durch „Memory Inhibit" sperrt und anstelle des Speichers die Daten auf den Bus legt. TA ist von der Funktion her vergleichbar mit STERM beim MC68030.
TEA (Transfer Error Acknowledge): Tritt ein Busfehler auf (Zugriff auf einen nicht vorhandenen Speicher), so wird TEA anstelle von TA aktiviert. Ist der MC68040 nicht Busmaster, wird TEA von der Snoop-Logik benutzt, um das Ende eines Buszyklusses festzustellen. TEA ist vergleichbar mit dem Buserror des MC68000.
TCI (Transfer Cache Inhibit): Ist dieses Signal bei einem Lesezyklus aktiv, so werden die durch diesen Zyklus übertragenen Daten nicht in das Cache abgelegt. Bei Burst-Transfers wird TCI nur beim ersten Langwort abgetastet.
TBI (Transfer Burst Inhibit): Ist eine angesprochene Einheit auf dem Bus nicht in der Lage, einen Burst Transfer durchzuführen, so wird durch das Transfer Burst Inhibit der Buscontroller dazu angewiesen, anstelle eines Burst-Zugriffes vier Langwortzugriffe durchzuführen. TBI wird nur beim ersten Langwort eines Bursts erkannt. Ist der Mikroprozessor MC68040 nicht Busmaster, wird TBI dazu verwendet, um das Ende eines jeden Buszyklus festzustellen.

Snoop-Kontrolle

SC0, SC1 (Snoop Control): Hiermit wird die Arbeitsweise der Snoop-Logik definiert. Sie werden dazu benutzt, die Snoop-Logik ein- bzw. auszuschalten. Die Bedeutung von SC0 und SC1 zeigt die folgende Tabelle für Lese- und Schreibzugriffe:

SC0	SC1	Lesezugriff	Schreibzugriff
0	0	kein Snooping	kein Snooping
0	1	"dirty" Daten ausgeben	Byte-, Wort- und Langwortdaten übernehmen
1	0	"dirty" Daten ausgeben und als ungültig markieren	Zeile für ungültig erklären
1	1	reserviert (kein Snooping)	reserviert (kein Snooping)

MI (Memory Inhibit): Wenn der MC68040 nicht Busmaster ist, wird dieses Ausgangssignal von der Snoop-Logik dazu benutzt, den Speicher abzuschalten. Ist die Snoop-Logik für einen Zugriff aktiv, so wird MI solange gesetzt, bis entschieden ist, ob der MC68040 aufgrund von inkonsistentem Cache anstelle des Speichers reagieren soll. Stellt die Snoop-Logik keine „dirty" Daten fest, wird MI inaktiv und der Speicher kann den Zyklus wie gewöhnlich beenden. Andernfalls bleibt MI aktiv, und der MC68040 beendet den Zugriff durch Aktivieren von TA. Dabei werden bei einem externen Lesezyklus die Daten aus dem Cache auf den Bus gelegt, bei einem Schreibzugriff werden die Daten vom Bus in das Cache übernommen.

Busarbitrierung

BR (Bus Request): Im Gegensatz zu den übrigen M68000-Mikroprozessoren ist BR beim MC68040 ein Ausgang. Benötigt der MC68040 den Bus, so aktiviert er BR.
BG (Bus Grant): Durch diesen Eingang informiert der Arbiter den MC68040 darüber, daß er den Bus benutzen darf, sobald der augenblickliche Master BB (Bus Busy) inaktiv ist und damit anzeigt, daß er den Bus freigegeben hat.
BB (Bus Busy): Ist der MC68040 nicht Busmaster, so überwacht er diesen Eingang um festzustellen, ob der Bus noch von anderen Einheiten benutzt wird. Der MC68040 übernimmt den Bus erst, wenn BG durch den Arbiter aktiviert wurde und BB inaktiv ist, also kein anderer Busmaster mehr arbeitet. Ist der MC68040 Busmaster, so aktiviert er BB und zeigt damit an, daß er den Bus benutzt.

Kontrollsignale

CDIS (Cache disable): Für Testzwecke können mit CDIS die Caches, ähnlich wie beim MC68020/68030, abgeschaltet werden. Die Cacheinhalte werden dabei nicht verändert und sind wieder gültig, sobald CDIS inaktiv wird. Die Snoop-Logik wird von CDIS nicht beeinflußt. Während

der RESET-Phase wird CDIS dazu benutzt, den Modus des Busses einzustellen: Ist CDIS beim RESET high, so benutzt der MC68040 anschließend den Bus in nicht gemuliplexter Form, ist CDIS beim RESET low, so wird der Bus gemultiplext.

MDIS (MMU disable): Für Testzwecke können damit die MMUs abgeschaltet werden, ähnlich wie durch MMUDIS beim MC68030. Einträge in den ACTs werden dabei nicht verändert und sind wieder gültig, sobald MDIS inaktiv wird.

RSTI (Reset In): RSTI versetzt den Prozessor in den RESET-Zustand. RSTI ist im Gegensatz zu den meisten übrigen Eingängen asynchron und wird intern auf die nächste steigende Flanke des Bustaktes synchronisiert.

RSTO (Reset Out): Bei der Ausführung des RESET-Befehls wird RSTO für 512 Bustakte aktiviert.

nen Busarbiter darüber zu informieren, den Bus an den Prozessor zu geben.

AVEC: Wird AVEC bei einem Interrupt-Acknowledge-Zyklus gleichzeitig mit TA aktiviert, so wird der MC68040 damit angewiesen, die entsprechenden, in der Vektortabelle definierten internen Auto-Interrupt-Vektor zu benutzen.

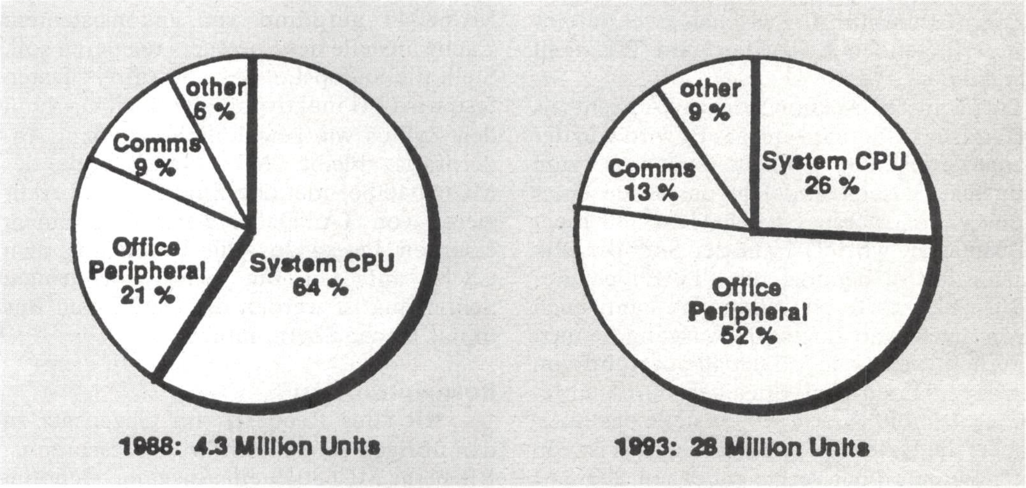
Statussignale

STT0-PST3 (Prozessor-Status): Diese Ausgänge zeigen den internen Zustand des MC68040 an und dienen dazu, interne Abläufe zu verfolgen, zum Beispiel für Debug-Zwecke. Sie haben allerdings keine weitere Bedeutung für den augenblicklichen Bustransfer.

Signale für Boundary Scan Test

TCK (Test Clock): Die Testlogik auf dem MC68040 bezieht ihren Takt aus die-

Der 32-bit-Mikroprozessor-Weltmarkt 1988 und 1993.



Interrupts

IPL0-IPL2 (Interrupt Priority Level): Diese drei Eingänge werden in gleicher Art und Weise verwendet wie bei den übrigen Mikroprozessoren aus der M68000-Familie. Sie sind low-aktiv und übergeben dem MC 68040 in binärer Form den Interrupt-Request-Level, der zwischen 0 (kein Interrupt), 1 (niedrigste Ebene) bis maximalen 7 (höchste Ebene, NMI) liegt.

Beim RESET werden die drei IPL-Eingänge dazu benutzt, die Charakteristik der Ausgangstreiber zu definieren. IPL2 definiert die Charakteristik der Datenbus-Treiber, IPL die der Audreßbuch-Treiber und IPL0 die der Treiber für die Kontrollsignale. Damit können die drei Teilbusse getrennt in Low-Drive- bzw. in High-Drive-Modus geschaltet werden.

IPEND (Interrupt Pending): Dieser Ausgang zeigt an, daß ein Interrupt intern erkannt worden ist, da seine Ebene höher ist, als die augenblickliche Interruptmaske im Statusregister. Das Signal ist identisch mit IPEND beim 68020/30 und dient dazu, ei-

sem Eingang. TCK ist völlig unabhängig vom Bus- oder Prozessortakt und kann damit gleichzeitig andere Komponenten versorgen.

TMS (Test Mode Select): Hiermit wird die Arbeitsweise der Testlogik festgelegt.

TDI (Test Data In): Die seriellen Testdaten werden über diesen Pin eingegeben.

TDO (Test Data Out): Die Ausgangsdaten der Testlogik werden seriell ausgegeben.

TRST (Test Reset): Damit wird die Testschaltung zurückgesetzt.

Zusammenfassung

Insgesamt führen die Optimierungen des MC68040 dazu, daß er sich von der Leistung her mit modernsten RISC-Prozessoren messen kann. Daneben bietet er volle Kompatibilität zur gesamten M68000-Familie. Damit steht eine riesige Menge an Anwendungs- und Systemsoftware für den MC68040 zur Verfügung. Ein Upgrade bestehender Systeme, die auf der M68000-Linie basieren, kann die bestehende Software problemlos nutzen.

Ein leistungsstarkes 32-bit-Minimal-system

David McCartney – Tommy Kelly

Zwar ist der 68040-Prozessor nun vorgestellt worden, dennoch bleiben 68030-Applikationen wegen ihrer Leistungsfähigkeit nach wie vor für die Anwender interessant.

Die vorliegende Applikationsschrift befaßt sich mit der Konstruktion eines leistungsfähigen 32-bit-Minimalsystems auf der Grundlage des 32-bit-Mikroprozessors MC68030 von Motorola. Die schnelle synchrone Schnittstelle dieses Prozessors wird hier genutzt, um Zugriffe auf den Schreib-Lesespeicher des Systems in nur zwei Taktzyklen abzuwickeln. Das System ist mit schnellen statischen RAM-Bausteinen bestückt und für eine Taktfrequenz von 20 MHz ausgelegt, kann aber mit schnelleren Speichern auf 25 beziehungsweise 30 MHz aufgerüstet werden.

Ein DUART (Dual-Channel Universal Asynchronous Receiver Transmitter) des Typs MC68681 implementiert mit seinen Zusatzschaltungen zwei serielle RS232-Ports, die für den Anschluß von Terminals beziehungsweise eines Hostcomputers dienen. Hinzu kommt ein PI/T (Parallel Interface/Timer) MC68230, der als parallele Centronics-Schnittstelle oder für allgemeine I/O-Aufgaben eingesetzt werden kann. Der 16-bit-Zeitgeber des MC68681 und der 24-bit-Timer im MC68230 stehen für allgemeine Zeitgeberfunktionen zur Verfügung.

Das Speichersystem gliedert sich in einen RAM-Teil aus acht schnellen 64-Kbit-SRAMs vom Typ MCM6164 und einen EPROM-Teil, der mit zwei 512-Kbit-EPROMs des Typs 27512 bestückt ist und zur Speicherung von Debug-Monitor-Firmware oder anderer Systemsoftware genutzt werden kann. Im Interesse möglichst hoher Leistung besitzt der RAM-Teil einen 32 bit breiten Datenbus, während der EPROM-Speicher mit einem 8-bit-Datenbus konfiguriert ist. Voraussetzung für diese unterschiedliche Auslegung ist die dynamische Busstruktur des MC68030.

Funktionsbeschreibung

Die RAM-Schnittstelle stellt den zeitkritischsten Abschnitt des gesamten Systems dar. Die Konstruktion wurde in allen Belangen auf die Zeitparameter dieser Schnittstelle hin optimiert, damit RAM-Zugriffe ohne Wartezyklen (Wait-States) möglich sind. Dies hat sich insbesondere auf die Konstruktion des Adreßdecoders ausgewirkt, weil das Handshakesignal STERM (Synchronous Termination) vor dem Beginn von Zustand 2 des Buszyklus am Prozessor anliegen muß, um einen Betrieb ohne Wartezyklen zu gewährleisten. Der Anschluß der EPROMs und der I/O-Funktionen wirkt sich auf die Systemleistung nicht so stark aus, so daß hier mit Wait-States und Zwischenpufferung gearbeitet werden kann.

Der Adreßdecoder

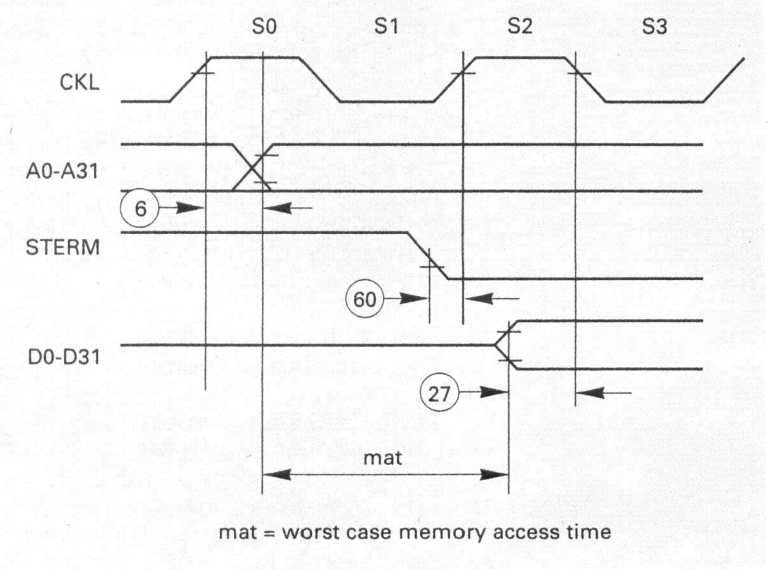
Aus einer Untersuchung des Zeitdiagramms für einen synchronen Lesezyklus des MC68030 (Bild unten) gehen folgende Kriterien hervor:

1. Die Adreßleitungen A0 bis A31, die Funktionscode-Leitungen FC0 bis FC2 sowie SIZ0 und SIZ1 werden an dem mit Punkt 6 bezeichneten Zeitpunkt nach der steigenden Flanke von Zustand 1 gültig.
2. Um einen Betrieb ohne Wartezyklen zu gewährleisten, muß dem synchronen Handshakesignal STERM vor der steigenden Flanke von Zustand 2 die mit Punkt 60 bezeichnete Vorbereitungszeit eingeräumt werden.

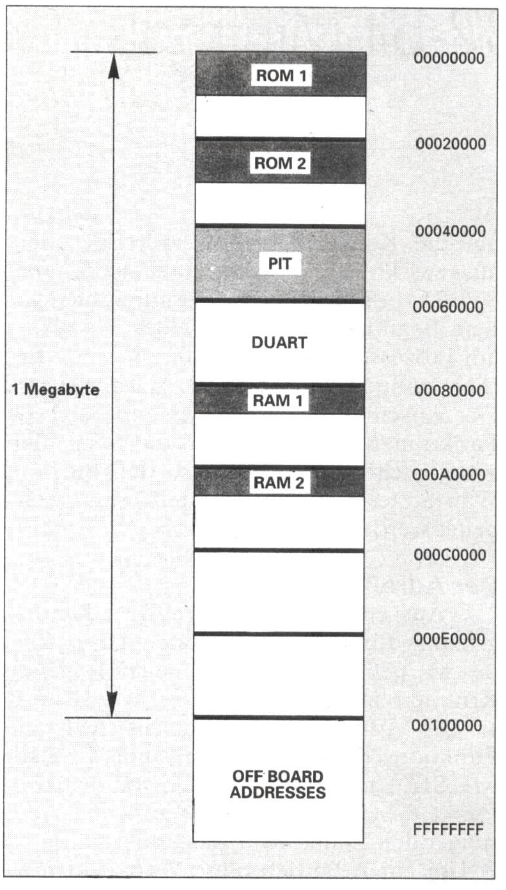
Somit steht das Intervall zwischen dem Maximalwert von Punkt 6 und dem Minimalwert von Punkt 60 zur Verfügung, um die Adresse zu decodieren und das STERM-Signal zu erzeugen. Ein mit 20 MHz getakteter MC68030 hat eine Zykluszeit von 50 ns. Somit sind für die Adreßde-

Die Autoren David McCartney und Tommy Kelly arbeiten als Entwicklungsingenieure bei Motorola in East Kilbridge, Schottland.

Zeitdiagramm für den synchronen Lesezyklus des MC68030



Adreßbelegung des MC68030-Systems



codierung und das Erzeugen des Handshakesignals Punkt 50 – Punkt min60 – Punkt max6 = 50 – 4 - 25 = 21 ns verfügbar.

In dieser Anwendung wird die Decodierung von drei 74F521- und zwei 74F138-Bausteinen (U14, U15 und U16 sowie U12 und U13) ausgeführt. Aus den Adreßleitungen A31 bis A24 wird das Decodiersignal BOARD_SEL erzeugt, mit dem das erste Megabyte des MC68030-Adreßbereichs dieser Applikationsplatine zugeordnet wird. Die Karte kann somit auch mit anderen Modulen verbunden werden, die den Speicherbereich weiter ausbaut. U15 und U16 erzeugen aus A23..A19 zwei Select-Signale, nämlich I/O.ROM_SEL für die unteren 12 KByte des Speicherbereichs und RAM_SEL für die folgenden 12 KByte. Diese beiden Select-Informationen werden durch U12 und U13 weiter decodiert, um die einzelnen Select-Signale für vier 128-K-RAM-Banks, zwei separate 128-KByte-ROM-Selects und zwei separate 128-KByte-I/O-Selects für den MC68681 und den MC68230 zu generieren. Das Bild oben zeigt die Speicherbelegung des Systems.

Das STERM-Signal wird in U8c durch Oder-Verknüpfung von BOARD_SEL und RAM_SEL gewonnen. Mit Hilfe der Drahtbrücke J6 ist es möglich, den Ausgang dieses Gatters direkt an den Prozessor zu führen oder ein System-STERM von außerhalb der Karte auf dem Weg über U7b zu übernehmen.

Mit U35 ist ein Wait-State-Generator realisiert worden, mit dem – selektiert durch J1 – zwischen 0 und 4 Wartezyklen erzeugt werden können. Die beschriebene Decodier- und Handshake-Logik erfüllt die zuvor aufgestellten Kriterien, denn der aus einem 74F521 und einem 74F32 gebildete Signalweg bewirkt eine maximale Signallaufzeit von 11 + 6,3 ns. Dieser Wert aber liegt deutlich unter der für den Betrieb ohne Wartezyklen notwendigen Grenze von 21 ns.

Der Festwertspeicher

Die beiden zur Aufnahme von EPROMs vorgesehenen Stecksockel des Systems sind für Bausteine des Typs 27512 konfiguriert. Die beiden EPROMs U25 und U26 sind mit den Bits D31..D24 des Datenbusses vom Prozessor verbunden, so daß der Datenport 8 bit breit ist. Dies hat den Vorzug, daß das Bootstrapping des Prozessors aus einem EPROM möglich ist. Die dynamische Busstruktur des MC68030 sorgt außerdem dafür, daß Zugriffe auf das Programm-ROM in jeder Breite automatisch korrekt ausgeführt werden.

Mit den beiden vom Adreßdecoder kommenden Chip-Select-Signalen ROM1CS und ROM2CS werden die beiden Bausteine selektiert. Weil das Interface 8 bit breit ist, müssen an den Prozessor die Handshake-Signale DSACK0 = 0 und DSACK1 = 1 übertragen werden. Die Zugriffszeit auf das ROM ist unkritisch, denn die meisten Anwenderprogramme werden aus dem RAM heraus verarbeitet. Es ist deshalb nicht nötig, die Anzahl der bei jedem EPROM-Zugriff entstehenden Wartezyklen zu reduzieren. Die beiden ROM-Chip-Select-Signale werden mit U7c in einer Und-Funktion verknüpft, das Resultat wiederum mit einem Verzögerungssignal in einem Oder-Glied verknüpft, um das erforderliche DSACK0-Signal für den Prozessor zu erzeugen. Das Verzögerungssignal liefert ein mit U4 realisierter zweiter Wartezyklus-Generator. Es entsteht eine Verzögerung von zwei Zyklen eines Taktes, dessen Frequenz ein Achtel des Prozessortaktes beträgt. Auf diese Weise wird selbst dann genügend Zeit für die EPROM-Zugriffe eingeräumt, wenn die Taktfrequenz des Prozessors höher gewählt wird als die in der Schaltung zugrunde gelegten 20 MHz.

Mit Hilfe des mit U4 implementierten Wait-State-Generators wird außerdem ein Time-out-Generator zum Abbruch fehlerhafter Zugriffe eingerichtet. Erhält der Prozessor nicht spätestens innerhalb von vier Zyklen – und zwar des langsamsten Takts im System – ein STERM-Signal oder das notwendige DSACKx-Handshake, so wird

am 68030-Mikroprozessor das BERR-Signal gesetzt, um die Exception-Verarbeitung aufzunehmen.

Schreib-/Lesespeicher

Wie schon erwähnt, erzeugt die Adreßdecoder-Logik vier separate RAM-Chip-Selects, die jeweils einen Bereich von 128 K ansprechen. Wegen der kapazitiven Kopplung durch die RAM-Chips wurde die Karte lediglich mit zwei Banks ausgestattet. Diese sind hier mit 45-ns-Bausteinen des Typs MCM6164 bestückt, die jedoch durch 32Kx8-Versionen ersetzt werden, sobald schnellere Speicher verfügbar sind. Weil die Decodier- und die Zugriffszeiten für diese RAMs zeitkritische Zeiten für das System sind, wurden sie für den Betrieb ohne Wartezyklen optimiert. Der RAM-Bereich ist mit einem 32-bit-Port ausgestattet und nutzt die im MC68030 neu eingeführte synchrone Schnittstelle aus. Damit lassen sich Lese- und Schreibzugriffe in je zwei Taktzyklen ausführen.

Um die Zugriffszeit weiter zu verbessern, besitzen die verwendeten RAM-Bausteine separate Chipfreigabe- und Ausgangsfreigabe-Anschlüsse. Die Chipfreigabe-Eingänge liegen fest auf logischem Null-Status, so daß sich die Chips ständig im Enable-Zustand befinden. Demgegenüber sind die Ausgangsfreigabe-Pins mit den vom Adreßdecoder erzeugten Chip-Select-Signalen verbunden. Bei einem Lesezugriff werden sämtliche vier Chips einer Bank gleichzeitig freigegeben, so daß die gesamten 32 bit auf den Datenbus gelegt werden. Der Prozessor wertet entsprechend der tatsächlichen Breite des jeweiligen Zugriffs nur die entsprechenden Teile des Datenbusses aus. Komplexer gestaltet sich ein vom Prozessor ausgeführter Schreibzugriff, weil nur die relevanten Teile des Datenbusses gültige Daten aufweisen. Das PAL U38 generiert aus den Signalen A0, A1, SIZ0 und SIZ1 des MC68030 die vier separaten Byte-Select-Signale UUD*, UMD*, LMD* und LLD*. In U39, U40, U41 und U42 werden diese Signale mit den Write-Select-Signalen RAM1 und RAM2 und dem Adreß-Strobe verknüpft, um die Schreibfreigabesignale für die einzelnen

U1	MC68030	32 bit CPU
U2	40 MHz xtal osc. mod.	
U3 - U4	74LS93	4 bit Counter
U5	74LS148	8 - 3 bit priority Encoder
U6	74LS138	3 - 8 Decoder
U7	74F08	Quad 2 input AND gate
U8	74F32	Quad 2 input OR gate
U9	74F04	Hex Inverter
U10	74F11	Triple 3 input AND gate
U11	74F20	Dual 4 input NAND gate
U12 - U13	74F138	3 - 8 Decoder
U14 - U16	74F521	8 bit Identity Comparator
U17 - U24	MCM6164	8K x 8 Static RAM 45ns
U25 - U26	27512	64K x 8 EPROM
U27	MC68230	Parallel Interface Timer (PI/T)
U28	MC68681	Dual UART (DUART)
U29	74LS244	Octal Buffer
U30 - U31	74F244	Octal Buffer
U32	NE555	Universal Timer
U33	74LS00	Quad 2 input NAND gate
U34	74LS05	Hex Inverter (Open Collector)
U35	74F175	Quad D type Flip Flop
U36 - U37	MAX232	+5v powered Dual RS232 Trans./Rec.
U38	PAL16L8	Programmable Logic Array
U39 - U42	74F32	Quad 2 input OR gate
J1		5 way jumper
J2		2 way jumper
J3		50 pin IDC connector
J4 - J5		DB - 25 connectors
J6 - J9		2 way jumper
S1 - S2		SPST Pushbutton

Bauteileliste

RAM-Bausteine zu gewinnen. Die zum Programmieren des PALs U38 benutzten Gleichungen sind dem Abschnitt „Applications" im Benutzerhandbuch zum MC68030 entnommen.

Um keine zusätzlichen Signallaufzeiten zu erzeugen, wird der Datenbus zwischen den RAM-Bausteinen und dem Prozessor nicht gepuffert. Der Adreßdecoder liefert zusätzliche RAM-Select-Signale für die Banks drei und vier, die der Karte mit einem gepufferten Adreß- und Datenbus hinzugefügt werden können.

Serielle Kommunikation

Der DUART MC68681 (U28) stellt zwei serielle Kommunikationskanäle zur Verfügung, die im allgemeinen zum Anschluß von lokalen Terminals oder eines entsprechenden Hostcomputers genutzt werden. Die beiden Treiberchips U36 und U37 dienen der zwischen dem DUART und den seriellen Ports erforderlichen Umsetzung zwischen TTL- und RS232-Pegeln. Herausgeführt sind die beiden Ports an den beiden DB25-Steckverbindern J4 und J5.

Der ebenfalls im DUART MC68681 enthaltene 16-bit-Zeitgeber kann im System für beliebige allgemeine Zähl- und Timeraufgaben herangezogen werden.

Paralleler Druckerport

Mit dem PI/T MC68230 (U27) verfügt das System über ein paralleles Centronics-Interface zum Anschluß eines Druckers. Der Ausgang des PI/T ist – gepuffert von einem Achtfach-Treiber 74LS244 (U29) – an einem 50poligen Standard-Steckverbinder (J3) herausgeführt. Der im MC68230 integrierte 24-bit-Zeitgeber kann unter anderem vom Betriebssystem zur Erzeugung periodischer Hardware-Interrupts verwendet werden.

Die Pufferung des Busses

Für den Betrieb ohne Wartezyklen sind die zeitkritischen Signalwege zum RAM ohne Puffer angelegt. An den Adreßleitungen A0..A15 liegen jedoch zusätzlich zwei 74F244 (U30 und U31). Die so gepufferten Leitungen sind mit den EPROMs und I/O-Funktionen des Systems verbunden, könnten jedoch ebenso für zusätzliche RAM-Banks eingesetzt werden. Im Datenbus befinden sich keinerlei Puffer. Vier bidirektionale Pufferbausteine des Typs 74F245 können hinzugefügt werden, wenn zusätzlicher RAM-Speicher benötigt wird.

Fazit

Diese Applikationsschrift beschreibt die Konstruktion eines leistungsstarken 32-bit-Systems, das die Eigenschaften der synchronen Busschnittstelle des 32-bit-Mikroprozessors MC68030 ausnutzt. Das System ist ursprünglich für eine Taktfrequenz von 20 MHz und (mit schnelleren Speichern) für den späteren Umbau auf 25 MHz ausgelegt. Um die Solidität der Konstruktion zu prüfen, wurde das System mit 30 MHz betrieben, in diesem Fall mit einem Wartezyklus für die RAM-Zugriffe bei Verwendung der normalen 45-ns-RAMs. Demonstriert wurde hierbei auch die Verwendung der synchronen Schnittstelle in Verbindung mit handelsüblichen Speichern und normaler FAST-TTL-Interfacelogik im Adreßdecoder.

Literatur: MC68030 Enhanced 32-bit Microprocessor User's Manual Second Edition MC68030UM/AD REV 1

Impressum
CHIP PLUS erscheint monatlich als Beilage zu CHIP.

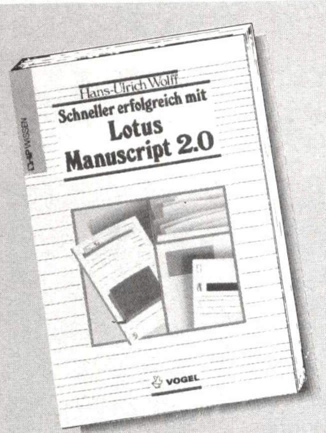
Anschrift der Redaktion: Schillerstr. 23a, 8000 München 2, Tel. 089-51493-0.
Verlag: Vogel Verlag und Druck KG, Max-Planck-Str. 7/9, 8700 Würzburg, Tel. 0931-418-0.
Redaktionsdirektor: Dr. Gerald O. Dick.
Redaktion: Martin Stübs (verantw.), Ulrich Parthier.
Chef vom Dienst: Reiner Lanninger.
Grafische Gestaltung: Hans Kuh, Nadja Schubert.

Textchef: Dr. Alfred Günter Neumann.
Autoren dieser Ausgabe: Josef Fuchs, David McCartney, Tommy Kelly.
Verlagsdirektor: Dr. Andreas Kaiser.
Anzeigen: Friedrich Mangold.
Satz, Litho, Druck, Verarbeitung, Versand: Vogel Verlag und Druck KG.

Der Verlag haftet nicht für unverlangt eingesandte Manuskripte und Fotos.

CHIP WISSEN

Die kompetente Buchreihe rund um den PC

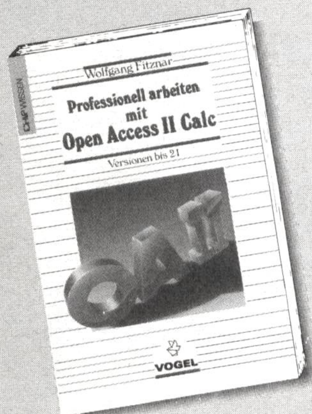


Hans-Ulrich Wolff:

Schneller erfolgreich mit Lotus Manuscript 2.0

192 Seiten, 82 Bilder
Hardcover, Format 24 × 16
40,– DM/ISBN 3-8023-**0222**-2

Dieses Buch gibt eine systematische Einführung in eine leistungsfähige Dokumentationsverarbeitung. Es beschreibt in einer systematischen Einführung die Programmbefehle beispielhaft Schritt für Schritt. Die detaillierte Gliederung sowie das umfangreiche Stichwortverzeichnis ermöglichen dem engagierten und geübteren Anwender das rasche Nachschlagen der ihn speziell interessierenden Abschnitte. Das Erzeugen hochwertiger Ausdrucke mit Grafiken, Tabellen und Gleichungen macht seinen Einsatz vor allem für technisch-wissenschaftliche Anwendungen interessant.



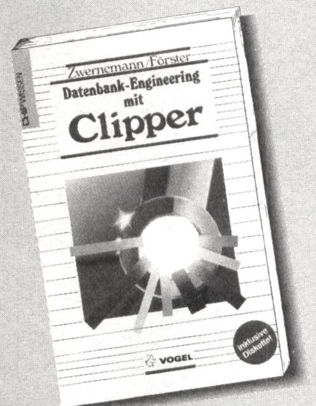
Wolfgang Fitznar:

Professionell arbeiten mit Open Access II Calc

Versionen bis 2.1

340 Seiten, 103 Bilder, 89 Tabellen
Hardcover, Format 24 × 16
48,– DM/ISBN 3-8023-**0298**-2

Dieses Buch unterstützt und begleitet den Anwender beim Einarbeiten in das Kalkulationsmodul »Calc« und die allgemeinen Bestandteile von Open Access II unter Berücksichtigung der neuen Version 2.1.
Aus dem Inhalt:
★ viele leicht nachvollziehbare und praxisnahe Beispiele ★ beigelegte Tastaturschablonen ★ aufwendige Grafiken ★ fundierte Besprechungen zu Kalkulationsmakros.
Der durchdachte Aufbau macht das Buch in Schulungen wie zum Selbststudium einsetzbar.

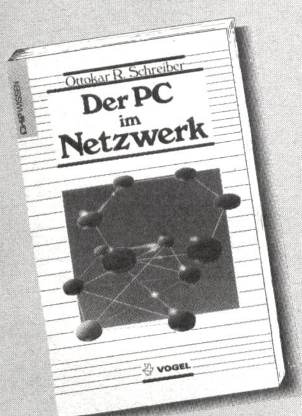


Zwernemann/Förster:

Datenbank-Engineering mit Clipper

464 Seiten, 58 Bilder, Hardcover
68,– DM **(inkl. 5,25-Zoll-Diskette)**
Format 24 × 16, ISBN 3-8023-**0263**-X

Dieses Buch beschreibt die Engineering-Praxis mit Clipper, welches **das** Entwicklungsprogramm für Datenbanksysteme ist. Der Inhalt konzentriert sich auf die Programmplanung, Entwicklung und Realisierung, wie sie im Rahmen eines kleinen Entwicklungsteams oder eines einzelnen Programmierers erfolgen sollte.
Typische Anwender sind mittelständische Unternehmen und Kleinbetriebe. Hierfür wurde ein konkretes Praxisbeispiel (ein komplettes Adreßverwaltungsprogramm) erarbeitet. Auf der beigefügten **5,25-Zoll-Diskette** finden Sie das komplette Datenbankprogramm, Checkliste und Planungsunterlagen.



Ottokar R. Schreiber:

Der PC im Netzwerk

224 Seiten, 72 Bilder, 18 Tabellen
Hardcover, Format 24 × 16
40,– DM/ISBN 3-8023-**0262**-1

Dieses Buch bietet fachliches Grundwissen über die Vernetzung und stellt schwierige Sachverhalte verständlich dar. Umfassende Informationssysteme, zunächst als Insellösungen in Form lokaler Netzwerke, halten Einzug in Büros, Unis usw. Es entsteht die Forderung über Verbindung dieser Kommunikationsinseln über Gateways, mit Großrechnern sowie Daten- und Wissensbanken. Zahlreiche LANs verbinden sich zu einem Internet, in dem jeder Teilnehmer auf alle darin verfügbaren Ressourcen zugreifen kann. Also insgesamt ein interessantes Gebiet. Greifen Sie zu!!

VOGEL

Haben Sie schon den neuen „CHIP- Katalog“? Bestellen Sie gleich!

Vogel Buchverlag
Postfach 67 40 · 8700 Würzburg 1

CHIP SPECIAL + CHIP WISSEN

Für Jeden das Richtige

Preisänderungen vorbehalten * mit 5,25" Diskette

▲ = Preis stand bei Drucklegung noch nicht fest

MS-DOS RECHNER

Wolf
DTP mit Ventura Publisher
ISBN 3-8023-0985-5, DM 28,-

Jordan
Unix lernen unter MS-DOS, Ausgabe 1
ISBN 3-8023-0988-7, DM 28,-

Ruprecht
UNIX-Einsatz auf dem PC, Ausgabe 2
ISBN 3-8023-0998-X, DM 28,-

Vollmuth
Laptop
ISBN 3-8023-1007-1, DM 28,-

CHIP SPECIAL Team
IBM PC, Kompatible, Ausg.2
Best.-Nr. 0070, DM 28,-

PC Public-Domain*
ISBN 3-8023-0970-7, DM 34,-

Mahnke
Projektmanagement mit dem PC
106 Seiten
ISBN 3-8023-0151-X, DM 30,-

Strauß
Sidekick Plus - kurz und bündig
180 Seiten
ISBN 3-8023-0401-2, DM 33,-

INTEGRIERTE SOFTWARE

Vollmuth
Open Access
Best.-Nr. 0520, DM 28,-

Fitznar
Prof. arbeiten mit Open Access II Calc
336 Seiten
ISBN 3-8023-0298-2, DM 48,-

Fitznar
Prof. arbeiten mit Open Access II Base
472 Seiten
ISBN 3-8023-0217-6, DM 68,-

Wolff
Schneller erfolgreich mit Framework III
180 Seiten
ISBN 3-8023-0287-7,
DM 38,-

SOFTWARE

CHIP SPECIAL Team
TDB-4 Die Turbo Datenbank mit ADL-System
ISBN 3-8023-0984-7, DM 99,-

DATENBANKEN

Parthier
Ingres-Datenbankmanagement
ISBN 3-8023-1012-8, DM 28,-

Herbert
dBase, Ausgabe 1
ISBN 3-8023-0981-2, DM 49,-

Herbert
Paradox
ISBN 3-8023-1009-8, DM 28,-

Zwernemann
Datenbank-Engineering mit Clipper*
464 Seiten
ISBN 3-8023-0263-X, DM 68,-

Beilstein
Professionell arbeiten mit Paradox*
500 Seiten
ISBN 3-8023-0255-9, DM 68,-

Schwinn
dBase IV kurz und bündig
ca.300 Seiten
ISBN 3-8023-0409-8,
ca. DM 43,- ▲

DATEN-ÜBERTRAGUNG

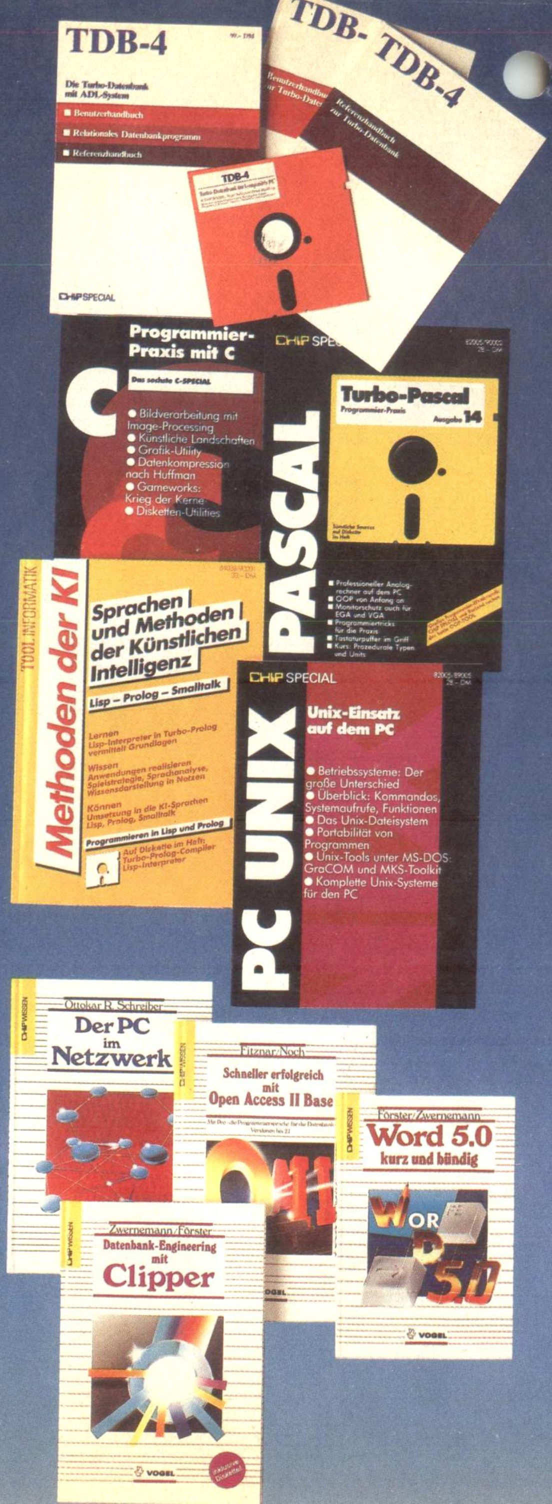
Müller / Zenk
Die neuen Wege lokaler Netze
ISBN 3-8023-1003-9, DM 49,-

Meder
OS/2 LAN Manager
ISBN 3-8023-1000-4, DM 28,-

Schreiber
Der PC im Netzwerk
224 Seiten
ISBN 3-8023-0262-1, DM 40,-

Jonas
Datenfernübertragung mit Personalcomputern
200 Seiten
ISBN 3-8023-0888-3, DM 40,-

Mußtopf
Mailbox-Praxis
ca. 160 Seiten
ISBN 3-8023-0253-2,
ca. DM 38,- ▲



BETRIEBSSYSTEME

Edlinger / Schwalm
MS-DOS, Ausgabe 1*
ISBN 3-8023-0949-9, DM 49,-

Edlinger / Schwalm
MS-DOS, Ausgabe 2*
ISBN 3-8023-0951-0, DM 49,-

Dereser
CP/M+
Best.-Nr. 0460, DM 28,-

Ruprecht
Mit UNIX in die Zukunft
ISBN 3-8023-0954-5, DM 28,-

Ruprecht
Unix nach dem X/Open-Standard*
ISBN 3-8023-1011-X, DM 28,-

Meder
OS/2 für jedermann
ISBN 3-8023-0982-0, DM 28,-

Biethan
MS-DOS / PC-DOS kurz und bündig
248 Seiten
ISBN 3-8023-0866-2, DM 38,-

Biethan
Systemprogrammierung unter MS-DOS / PC-DOS*
2.Aufl. 1990, ca. 500 S.
ISBN 3-8023-0178-1, DM 68,-

TEXT-VERARBEITUNG

Förster/Zwernemann
Word 5.0 – Layout und Druckform*
ca. 180 S.
ISBN 3-8023-0422-5,ca.DM 48,-

Förster
Rationelle Textverarbeitung mit Word-Perfect 5.0
ca. 160 Seiten
ISBN 3-8023-0295-8, DM 48,-

Förster
Word 5.0 kurz und bündig
ca. 272 Seiten
ISBN 3-8023-0418-7, DM 43,-

Langfelder
WordStar Professional kurz und bündig
ca. 200 Seiten
ISBN 3-8023-0296-6, DM 38,-

Wolff
Lotus Manuscript 2.0
152 Seiten
ISBN 3-8023-0222-2, DM 40,-

Förster
IBM PC Text 4 kurz und bündig
196 Seiten
ISBN 3-8023-0247, DM 38,-

PROGRAMMIER-SPRACHEN

Binder
C – Der schnelle Einstieg, Ausg. 1
Best.-Nr. 0440, DM 28,-

Binder
Professionell arbeiten mit Turbo-C, Ausg. 2
ISBN 3-8023-0964-2, DM 28,-

Binder
C Praxis, Ausg. 3
ISBN 3-8023-0976-6, DM 28,-

Knappe / Rahlff
C-Praxis, Ausg. 4
ISBN 3-8023-0991-X, DM 28,-

Knappe / Rahlff
C-Praxis, Ausg. 5
ISBN 3-8023-0995-2, DM 28,-

TOOL Praxis Team
Modula 2, Ausgabe 1
ISBN 3-8023-0992-8, DM 18,-

TOOL Praxis Team
Modula 2, Ausgabe 2
ISBN 3-8023-1015-2, DM 18,-

TOOL Praxis Team
Assembler
ISBN 3-8023-1014-4, DM 18,-

CHIP Professional
Programmieren, Ausgabe 7
ISBN 3-8023-0996-0, DM 19,-

CHIP Professional
Programmieren, Ausgabe 6
ISBN 3-8023-0993-6, DM 19,-

CHIP Professional
Programmieren, Ausgabe 5
ISBN 3-8023-0990-1, DM 19,-

CHIP Professional
Programmieren, Ausgabe 4
ISBN 3-8023-0987-1, DM 19,-

CHIP Professional
Programmieren, Ausgabe 3
ISBN 3-8023-0974-X, DM 19,-

TOOL Informatik Methoden der KI*
inkl. 5,25"-Diskette
ISBN 3-8023-0997-9, DM 28,-

TOOL Praxis Team Quick Basic
ISBN 3-8023-1016-0, DM 28,-

Sacht
Programmieren mit Quick BASIC 4.0 und 4.5*
210 Seiten
ISBN 3-8023-0245-1, DM 48,-

Biethan
Programmieren mit Quick C 2.0*
ca. 300 Seiten
ISBN 3-8023-0414-0,
ca.DM 53,-

Kern
Wie man in Modula 2 programmiert
256 Seiten
ISBN 3-8023-0207-9, DM 40,-

Kasper
Objektorientiertes Programmieren in Smalltalk*
268 Seiten
ISBN 3-8023-0237-9, DM 58

Reuner
Turbo Pascal 5.0/5.5*
ca. 324 Seiten
ISBN 3-8023-0415-2,
ca. DM 58,- ▲

DIE TURBO-SERIE

Kern / Ullrich u.a.
Das Allerbeste aus der Turbo-Serie*
Best.-Nr. 1008, DM 49,-

Turbo-Pascal, Ausgabe 8
ISBN 3-8023-0968-5, DM 28,-

Turbo-Pascal, Ausgabe 9
ISBN 3-8023-0972-3, DM 28,-

Turbo-Pascal, Ausgabe 10
ISBN 3-8023-0975-8, DM 28,-

Turbo-Pascal, Ausgabe 11
ISBN 3-8023-0978-2, DM 28,-

Turbo-Pascal, Ausgabe 12
ISBN 3-8023-0989-8, DM 28,-

Turbo-Pascal, Ausgabe 13
ISBN 3-8023-0994-4, DM 28,-

Turbo Pascal, Ausgabe 14*
ISBN 3-8023-0973-1, DM 28,-

Turbo-Basic, Ausgabe 1*
Best.-Nr. 0959, DM 28,-

Turbo-Prolog*
Best.-Nr. 0540, DM 28,-



• CHIP WISSEN